IMM

DTU

OPTIMIZATION WITH CONSTRAINTS

2nd Edition, March 2004

K. Madsen, H.B. Nielsen, O. Tingleff

Informatics and Mathematical Modelling
Technical University of Denmark

CONTENTS

1. INTRODUCTION

In this booklet we shall discuss numerical methods for constrained optimization problems. The description supplements the description of unconstrained optimization in Frandsen et al (1998). We consider a real function of a vector variable which is constrained to satisfy certain conditions, specifically a set of equality constraints and a set of inequality constraints.

Definition 1.1. Feasible region. A point $\mathbf{x} \in \mathbb{R}^n$ is *feasible* if it satisfies the *equality constraints*

$$c_i(\mathbf{x}) = 0, \;\; i = 1, \ldots, r, \qquad\qquad r \geq 0$$

and the *inequality constraints*

$$c_i(\mathbf{x}) \geq 0, \;\; i = r+1, \ldots, m, \qquad m \geq r \;,$$

where the $c_i : \mathbb{R}^n \mapsto \mathbb{R}$ are given.

The set of feasible points is denoted by $\mathcal{P}$ and called the *feasible region*.

Notice that if $r = 0$, then we have no equality constraints, and if $r = m$ we have no inequality constraints.

A constrained minimizer gives a minimal value of the function while satisfying all constraints, ie

Definition 1.2. Global constrained minimizer. Find

$$\mathbf{x}^+ \;=\; \text{argmin}_{\mathbf{x} \in \mathcal{P}} f(\mathbf{x}) \;,$$

where $f \;:\; \mathbb{R}^n \mapsto \mathbb{R}$ and $\mathcal{P}$ is given in Definition 1.1.

Here, f is the so-called *objective function* or *cost function*.

Example 1.1. In $\mathbb{R}^1$ consider the objective function $f(x) = (x-1)^2$. The *unconstrained minimizer* is $x_{\mathrm{u}}^+ = 1$ with $f(x_{\mathrm{u}}^+) = 0$. We shall look at the effect of some simple constraints.

1°. With the constraint $x \geq 0, \;\big(r = 0, \; m = 1, \; c_1(x) = x\big)$ we also find the constrained minimizer $x^+ = 1$.

2° With the constraint $x - 2 \geq 0$ the feasible region is the interval $\mathcal{P} = [2, \infty[$, and $x^+ = 2$ with $f(x^+) = 1$.

3° The inequality constraints given by $c_1(x) = x - 2$ and $c_2(x) = 3 - x$ lead to $\mathcal{P} = [2, 3]$ and $x^+ = 2$.

4° If we have the equality constraint $3 - x = 0$, then the feasible region consists of one point only, $\mathcal{P} = \{3\}$, and this point will be the minimizer.

5° Finally, $3 - x \geq 0, \;\; x - 4 \geq 0$ illustrates that $\mathcal{P}$ may be empty, in which case the constrained optimization problem has no solution. ∎

In many constrained problems the solution is at the border of the feasible region (as in cases 2° – 4° in Example 1.1). Thus a very important special case is the set of points in $\mathcal{P}$ which satisfy some of the inequality constraints to the limit, ie with equality. At such a point $\mathbf{z} \in \mathcal{P}$ the corresponding constraints are said to be *active*. For practical reasons a constraint which is not satisfied at $\mathbf{z}$ is also called active at $\mathbf{z}$.

Definition 1.3. Active constraints. A constraint $c_k(\mathbf{x}) \geq 0$ is said to be

$$\textit{active at } \mathbf{z} \in \mathbb{R}^n \quad \text{if } \; c_k(\mathbf{z}) \leq 0 \;,$$

$$\textit{inactive at } \mathbf{z} \in \mathbb{R}^n \quad \text{if } \; c_k(\mathbf{z}) > 0 \;.$$

The **active set** at $\mathbf{z}$, $\mathcal{A}(\mathbf{z})$, is the set of indices of equality constraints and active inequality constraints:

$$\mathcal{A}(\mathbf{z}) = \{1, \ldots, r\} \cup \widetilde{\mathcal{A}}(\mathbf{z}) \;,$$

$$\text{where} \quad \widetilde{\mathcal{A}}(\mathbf{z}) = \{j \in \{r+1, \ldots, m\} \mid c_j(\mathbf{z}) \leq 0\} \;.$$

Thus, an inequality constraint which is inactive at $\mathbf{z}$ has no influence on the optimization problem in a neighbourhood of $\mathbf{z}$.

Example 1.2. In case 3° of Example 1.1 the constraint c_1 is active and c_2 is inactive at the solution x^+. Here the active set is

$$\mathcal{A}(x^+) = \widetilde{\mathcal{A}}(x^+) = \{1\} \; . \qquad \blacksquare$$

As in unconstrained optimization a global, constrained minimizer (Definition 1.2) can only be computed under special circumstances, like for instance convexity of some of the functions. In some cases (including some non-convex problems) methods of interval analysis can be applied to find a global, constrained minimizer (see for instance Caprani et al (2002)).

In this booklet, however, we shall only discuss methods that determine a local constrained minimizer. Such a method provides a function value which is minimal inside a feasible neighbourhood, determined by ε ($\varepsilon > 0$) :

Definition 1.4. Local constrained minimizer. Find $\mathbf{x}^* \in \mathcal{P}$ so that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{P} \text{ with } \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon \; .$$

Theorem 1.5. Assume that

0) The feasible region $\mathcal{P}$ is not empty

1) c_i $(i = 1, \ldots, m)$ are continuous for all $\mathbf{x} \in \mathcal{P}$

2) f is continuous for all $\mathbf{x} \in \mathcal{P}$

3) $\mathcal{P}$ is bounded $(\exists\, C \in \mathbf{R} : \|\mathbf{x}\| \leq C \text{ for all } \mathbf{x} \in \mathcal{P})$

Then there exists (at least) one global, constrained minimizer.

If both the cost function f and all constraint functions c_i are linear in $\mathbf{x}$, then we have a so-called a *linear optimization problem*. The solution of such problems is treated in Nielsen (1999). In another important special case all constraints are linear, and f is a quadratic polynomial; this is called a *quadratic optimization problem*, see Chapter 3.

We conclude this introduction with two sections on important properties of the functions involved in our problems.

1.1. Smoothness and Descent Directions

In this booklet we assume that the cost function satisfies the following Taylor smoothness condition,

$$f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \mathbf{h}^\top \mathbf{g} + \tfrac{1}{2}\mathbf{h}^\top \mathbf{H}\, \mathbf{h} + O(\|\mathbf{h}\|^3) \tag{1.6a}$$

where $\mathbf{g}$ is the *gradient*,

$$\mathbf{g} \;\equiv\; \mathbf{f}'(\mathbf{x}) \;\equiv\; \begin{bmatrix} \dfrac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \dfrac{\partial f}{\partial x_n}(\mathbf{x}) \end{bmatrix} \; , \tag{1.6b}$$

and $\mathbf{H}$ is the *Hessian matrix*,

$$\mathbf{H} \;\equiv\; \mathbf{f}''(\mathbf{x}) \;\equiv\; \left[\{ \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}) \} \right] \; . \tag{1.6c}$$

Furthermore we assume that the feasible region $\mathcal{P}$ has a piecewise smooth boundary. Specifically we request the constraint functions to satisfy the following Taylor smoothness condition,

$$c_i(\mathbf{x}+\mathbf{h}) = c_i(\mathbf{x}) + \mathbf{h}^\top \mathbf{a}_i + \tfrac{1}{2}\mathbf{h}^\top \mathbf{A}_i \mathbf{h} + O(\|\mathbf{h}\|^3) \tag{1.7a}$$

for $i = 1, \ldots, m$. Here $\mathbf{a}_i$ and $\mathbf{A}_i$ represent the gradient and the Hessian matrix respectively,

$$\mathbf{a}_i = \mathbf{c}_i'(\mathbf{x}) \; , \qquad \mathbf{A}_i = \mathbf{c}_i''(\mathbf{x}) \; . \tag{1.7b}$$

Notice that even when (1.7) is true, the boundary of $\mathcal{P}$ may contain points, curves, surfaces (and other subspaces), where the boundary is not smooth, eg points where more than one inequality constraint is active.

Example 1.3. We consider a two-dimensional problem with two inequality constraints, $c_1(\mathbf{x}) \geq 0$ and $c_2(\mathbf{x}) \geq 0$.

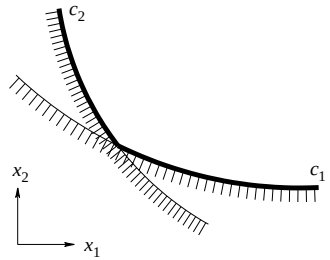


Figure 1.1: *Two inequality constraints in* $\mathbb{R}^2$.
The infeasible side is hatched.
In this and the following figures c_1 *means the set* $\{\mathbf{x} \mid c_1(\mathbf{x}) = 0\}$, *etc.*

In Figure 1.1 you see two curves with the points where c_1 and c_2, respectively, is active, see Definition 1.3. The infeasible side of each curve is indicated by hatching. The resulting boundary of $\mathcal{P}$ (shown with thick line) is not smooth at the point where both constraints are active. You can also see that at this point the tangents of the two "active curves" form an angle which is less than (or equal to) π, when measured inside the feasible region. This is a general property.

Next, we consider a three-dimensional problem with two inequality constraints. Below, you see the active surfaces $c_1(\mathbf{x}) = 0$ and $c_2(\mathbf{x}) = 0$. As in the 2-dimensional case we have marked the actual boundary of the feasible region by thick line and indicated the infeasible side of each constraint by hatching. It is seen that the intersection curve is a kink line in the boundary surface. It is also seen that the angle between the intersecting constraint surfaces is less than (or equal to) π, when measured inside $\mathcal{P}$.

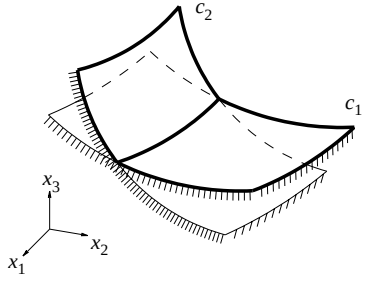


Figure 1.2: *Two inequality constraints in* $\mathbb{R}^3$.

■

The methods we present in this booklet are in essence *descent methods*, ie iterative methods where we move from the present position $\mathbf{x}$ in a direction $\mathbf{h}$ that provides a smaller value of the cost function. We must satisfy the *descent condition*

$$f(\mathbf{x}+\mathbf{h}) < f(\mathbf{x}) \,. \tag{1.8}$$

In Frandsen et al (1999) we have shown that the direction giving the fastest local descent rate is the *Steepest Descent* direction

$$\mathbf{h}_{\text{sd}} = -\mathbf{f}'(\mathbf{x}) \,. \tag{1.9}$$

In the same reference we also showed that the hyperplane

$$\mathcal{H}(\mathbf{x}) = \{\mathbf{x}+\mathbf{u} \mid \mathbf{u}^\top \mathbf{f}'(\mathbf{x}) = 0\} \tag{1.10}$$

divides the space $\mathbb{R}^n$ into a "descent" (or "downhill") half space and an "uphill" half space.

A descent direction $\mathbf{h}$ is characterized by having a positive projection onto the steepest descent direction,

$$\mathbf{h}^\top \mathbf{h}_{\text{sd}} > 0 \quad \Longleftrightarrow \quad \mathbf{h}^\top \mathbf{f}'(\mathbf{x}) < 0 \,. \tag{1.11}$$

Now consider the constraint functions. The equality constraints $c_i(\mathbf{x}) = 0 \ (i = 1, \ldots, r)$ and the boundary curves corresponding to the **active** inequality constraints, $c_i(\mathbf{x}) \geq 0$ satisfied with "=", can be considered as level curves or contours $(n = 2)$, respectively level surfaces or contour surfaces $(n > 2)$, for these functions. We truncate the Taylor series (1.7) to

$$c_i(\mathbf{x}+\mathbf{h}) = c_i(\mathbf{x}) + \mathbf{h}^\top \mathbf{a}_i + O(\|\mathbf{h}\|^2) \quad (i = 1, \ldots, m) \,.$$

From this it can be seen that the direction $\mathbf{a}_i \ (= \mathbf{c}_i'(\mathbf{x}))$ is orthogonal to any tangent to the contour at position $\mathbf{x}$, ie $\mathbf{a}_i$ is a normal to the constraint curve (surface) at the position.

Example 1.4. We continue the 2- and 3-dimensional considerations from Example 1.3, see Figure 1.3. At the position $\mathbf{x}$ c_1 is an active and c_2 is an inactive constraint, ie $c_1(\mathbf{x}) = 0$ and $c_2(\mathbf{x}) > 0$.

The opposite is true at the position $\mathbf{y}$, $c_2(\mathbf{y}) = 0$ and $c_1(\mathbf{y}) > 0$.

At the two positions we have indicated the gradients of the active constraints. They point into the interior of $\mathcal{P}$, the feasible region.

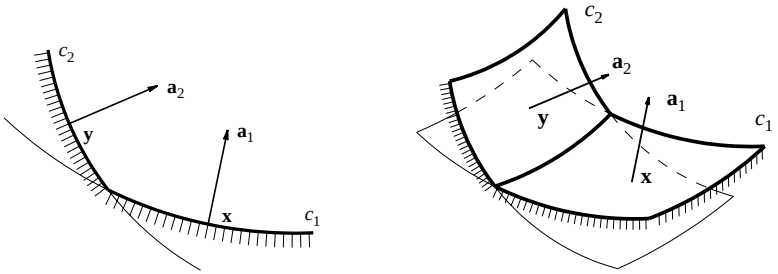


Figure 1.3: *The gradients of the constraint functions in* $\mathbf{R}^2$ *and* $\mathbf{R}^3$ *point into the feasible region.*

At this early stage we want to emphasize that in a neighbourhood of a point at the boundary of $\mathcal{P}$, properties of inactive constraints have no influence. ∎

1.2. Convexity

The last phenomenon to be described in this introduction is *convexity*. It is essential for a theorem on uniqueness of a constrained global minimizer and also for some special methods.

A set $\mathcal{D}$ (for instance the feasible region $\mathcal{P}$) is convex if the line segment between two arbitrary points in the set is contained in the set:

Definition 1.12. Convexity of a set. The set $\mathcal{D} \subseteq \mathbf{R}^n$ is convex if the following holds for arbitrary $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\theta \in [0,1]$ and $\mathbf{x}_\theta \equiv \theta\mathbf{x} + (1-\theta)\mathbf{y}$:

$$\mathbf{x}_\theta \in \mathcal{D} \ .$$

We also use the term convexity about functions:

Definition 1.13. Convexity of a function. Assume that $\mathcal{D} \subseteq \mathbf{R}^n$ is convex. The function f is *convex* on $\mathcal{D}$ if the following holds for arbitrary $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\theta \in [0,1]$ and $\mathbf{x}_\theta \equiv \theta\mathbf{x} + (1-\theta)\mathbf{y}$:

$$f(\mathbf{x}_\theta) \le \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \ .$$

f is *strictly convex* on $\mathcal{D}$ if

$$f(\mathbf{x}_\theta) < \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \ .$$

Definition 1.14. Concavity of a function. Assume that $\mathcal{D} \subseteq \mathbf{R}^n$ is convex. The function f is *concave/strictly concave* on $\mathcal{D}$ if $-f$ is convex/strictly convex on $\mathcal{D}$.

In the figure we show a strictly convex function between two points $\mathbf{x}, \mathbf{y} \in \mathcal{D}$. The definition says that $f(\mathbf{x}_\theta)$, with $\mathbf{x}_\theta \equiv \theta\mathbf{x} + (1-\theta)\mathbf{y}$, is below the secant between the points $\big(0, f(\mathbf{x})\big)$ and $\big(1, f(\mathbf{y})\big)$, and this holds for all choices of $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{D}$.

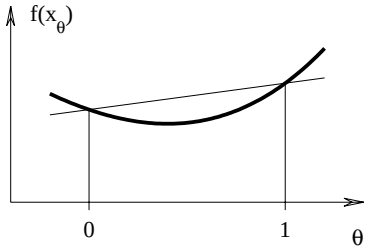


Figure 1.4: *A strictly convex function.*

Definition 1.15. Convexity at a point. The function f is *convex* at $\mathbf{x} \in \mathcal{D}$ if there exists $\epsilon > 0$ such that for arbitrary $\mathbf{y} \in \mathcal{D}$ with $\|\mathbf{x}-\mathbf{y}\| < \epsilon$, $\theta \in [0,1]$ and $\mathbf{x}_\theta \equiv \theta\mathbf{x} + (1-\theta)\mathbf{y}$:

$$f(\mathbf{x}_\theta) \le \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \ .$$

f is *strictly convex* at $\mathbf{x} \in \mathcal{D}$ if

$$f(\mathbf{x}_\theta) < \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \ .$$

It is easy to prove the following results:

Theorem 1.16. If $\mathcal{D} \subseteq \mathbf{R}^n$ is convex and f is twice differentiable on $\mathcal{D}$, then

1° f is convex on $\mathcal{D}$

$\iff$ $\mathbf{f}''(\mathbf{x})$ is positive semidefinite for all $\mathbf{x} \in \mathcal{D}$

2° f is strictly convex on $\mathcal{D}$ if

$\mathbf{f}''(\mathbf{x})$ is positive definite for all $\mathbf{x} \in \mathcal{D}$

Theorem 1.17. First sufficient condition. If $\mathcal{P}$ is bounded and convex and if f is convex on $\mathcal{P}$, then

f has a unique global minimizer in $\mathcal{P}$.

Theorem 1.18. If f is twice differentiable at $\mathbf{x} \in \mathcal{D}$, then

1° f is convex at $\mathbf{x} \in \mathcal{D}$

$\iff$ $\mathbf{f}''(\mathbf{x})$ is positive semidefinite

2° f is strictly convex at $\mathbf{x} \in \mathcal{D}$ if

$\mathbf{f}''(\mathbf{x})$ is positive definite

We finish this section with two interesting observations about the feasible domain $\mathcal{P}$.

1) Let c_i be an equality constraint. Take two arbitrary feasible points $\mathbf{x}$ and $\mathbf{y}$: $c_i(\mathbf{x}) = c_i(\mathbf{y}) = 0$. All points $\mathbf{x}_\theta$ on the line between $\mathbf{x}$ and $\mathbf{y}$ must also be feasible (cf Definition 1.12),

$$c_i(\theta\mathbf{x} + (1-\theta)\mathbf{y}) = 0 \quad \text{for all } \theta \in [0,1] .$$

Thus c_i must be linear, and we obtain the surprising result: If $\mathcal{P}$ is convex, then all equality constraints are linear. On the other hand the set of points satisfying a linear equality constraint must be convex. Therefore the feasible domain of an equality constrained problem (ie $r = m$) is convex if and only if all constraints are linear.

2) Let c_i be an inequality constraint. Assume that c_i is concave on $\mathbf{R}^n$. If $c_i(\mathbf{x}) \geq 0$, $c_i(\mathbf{y}) \geq 0$ and $\theta \in [0,1]$, then Definition 1.14 implies that

$$c_i(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \geq \theta\, c_i(\mathbf{x}) + (1-\theta)\, c_i(\mathbf{y})) \geq 0 .$$

Thus, the set of points where c_i is satisfied is a convex set. This means that the feasible domain $\mathcal{P}$ is convex if all equality constraints are linear and all inequality constraints are concave.

2. LOCAL, CONSTRAINED MINIMIZERS

We shall progress gradually when we introduce the different aspects and complications in the conditions for local, constrained minimizers (see Definition 1.4). We make the assumption that the feasible region $\mathcal{P}$ is not empty and that the cost function f and the constraint functions c_i $(i=1,\ldots,m)$ are smooth enough for the Taylor expansions (1.6) and (1.7) to hold.

First, we consider some special cases.

0) No equality constraints, no active inequality constraints

In Figure 2.1 we indicate the current position $\mathbf{x}$ and the "downhill" halfspace (cf (1.9) and (1.10)) in the two-dimensional case. The "uphill" side of the dividing hyperplane $\mathcal{H}$ is hatched.

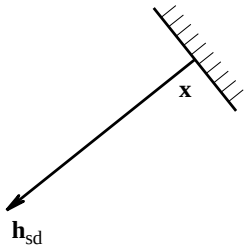


Figure 2.1: *Steepest descent direction and "downhill" halfspace.*

In order to get a lower cost value we should move in a direction $\mathbf{h}$ in the unhatched halfspace of descent directions. If the step is not too long then the constraints of the problem are of no consequence.

1) One equality constraint (no inequality constraints) in $\mathbf{R}^2$

In this case the feasible region, $\mathcal{P} = \{\mathbf{x} \mid c_1(\mathbf{x}) = 0\}$, is the curve shown below. At a position $\mathbf{x}$ we show $\mathbf{h}_{\text{sd}}$, the halfspace of descent directions, and the constraint gradient, $\mathbf{a}_1 = \mathbf{c}_1'(\mathbf{x})$, see (1.7b). It is obvious that if $\mathbf{x}$ "slides to the left" along the constraint curve, "pulled by the force $\mathbf{h}_{\text{sd}}$", we shall get lower cost values and still remain feasible. Thus $\mathbf{x}$ cannot be a local, constrained minimizer.

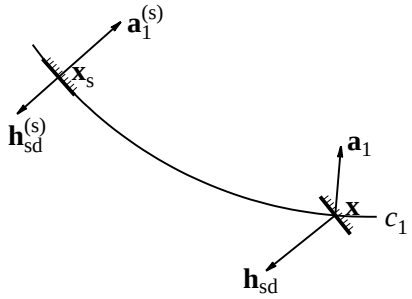


Figure 2.2: *One equality constraint in* $\mathbf{R}^2$, $c_1(\mathbf{x}) = 0$.
At $\mathbf{x}$ *we have a feasible descent direction; at* $\mathbf{x}_s$ *there is none.*

At the position $\mathbf{x}_s$ the vectors $\mathbf{h}_{\text{sd}}^{(s)} = -\mathbf{f}'(\mathbf{x}_s)$ and $\mathbf{a}_1^{(s)} = \mathbf{c}_1'(\mathbf{x}_s)$ are proportional, ie they satisfy a relation of the form

$$\mathbf{h}_{\text{sd}}^{(s)} = -\lambda\,\mathbf{a}_1^{(s)} \quad \Longleftrightarrow \quad \mathbf{f}'(\mathbf{x}_s) = \lambda\,\mathbf{c}_1'(\mathbf{x}_s)\,, \tag{2.1}$$

where λ is a scalar. This point may be a local, constrained minimizer. We say that $\mathbf{x}_s$ is a *constrained stationary point*.

In conclusion: Any local, constrained minimizer must satisfy the equation in (2.1) with $\lambda \in \mathbf{R}$.

2) Two active inequality constraints (no equality constraints) in $\mathbf{R}^2$

Figure 2.5 illustrates this case. At position $\mathbf{x}$ both constraints are active. The pulling force $\mathbf{h}_{\text{sd}}$ shown indicates that the entire feasible region is on the ascent side of the dividing plane $\mathcal{H}$ (defined in (1.10) and indicated in Figure 2.5 by a dashed line). In this case, $\mathbf{x}$ is a local, constrained minimizer.

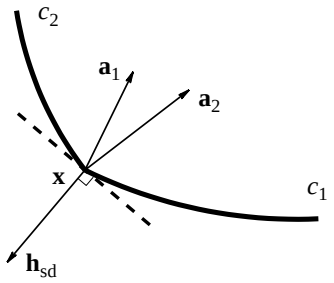


Figure 2.5: *Two inequality constraints in* $\mathbb{R}^2$, $c_1(\mathbf{x}) \geq 0$ *and* $c_2(\mathbf{x}) \geq 0$. *At the intersection point,* $\mathbf{h}_{sd}$ *points out of the feasible region.*

Imagine that you turn $\mathbf{h}_{\text{sd}}$ around the point $\mathbf{x}$ (ie, you change the cost function f). As soon as the dividing plane intersects with the active part of one of the borders, a feasible descent direction appears. The limiting cases are, when $\mathbf{h}_{\text{sd}}$ is opposite to either $\mathbf{a}_1$ or $\mathbf{a}_2$. The position $\mathbf{x}_{\text{s}}$ is said to be a *constrained stationary point* if $\mathbf{h}_{\text{sd}}^{(\text{s})}$ is inside the angle formed by $-\mathbf{a}_1$ and $-\mathbf{a}_2$, or

$$\mathbf{h}_{\text{sd}}^{(\text{s})} = -\lambda_1 \mathbf{a}_1^{(\text{s})} - \lambda_2 \mathbf{a}_2^{(\text{s})} \quad \text{with } \lambda_1, \lambda_2 \geq 0 \, .$$

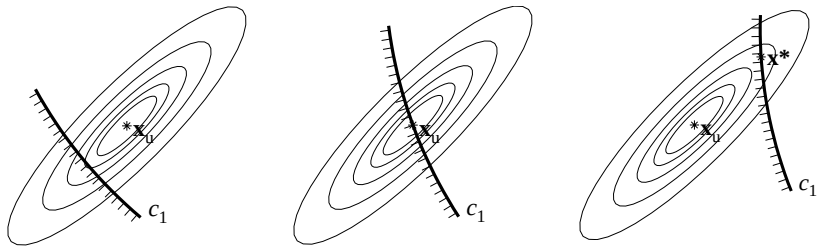
This is equivalent to

$$\mathbf{f}'(\mathbf{x}_{\text{s}}) = \lambda_1 \mathbf{c}_1'(\mathbf{x}_{\text{s}}) + \lambda_2 \mathbf{c}_2'(\mathbf{x}_{\text{s}}) \quad \text{with } \lambda_1, \lambda_2 \geq 0 \, . \tag{2.2}$$

3) Strongly and weakly active constraints (in $\mathbb{R}^2$*)*

In Figure 2.6 we show one inequality constraint and the contours of a quadratic function together with $\mathbf{x}_{\text{u}}$, its unconstrained minimizer. In the first case, $\mathbf{x}_{\text{u}}$ is also a constrained minimizer, because the constraint c_1 is inactive. In the last case, $\mathbf{x}_{\text{u}}$ is not feasible and we have a constrained minimizer $\mathbf{x}^*$, with $\mathbf{f}'(\mathbf{x}^*) \neq \mathbf{0}$. We say that the constraint is *strongly active*.

In the middle case, c_1 is active at $\mathbf{x}_u$. Here we say that the constraint is *weakly active*, corresponding to $\lambda_1 = 0$ in (2.1) and (2.2) (because $\mathbf{f}'(\mathbf{x}_u) = \mathbf{0}$).



c_1 *is inactive* c_1 *is weakly active* c_1 *is strongly active*

Figure 2.6: *Contours of a quadratic function in* $\mathbb{R}^2$*; one constraint,* $c_1(\mathbf{x}) \geq 0$.

In other words: if a constraint is weakly active, we can discard it without changing the optimizer. This remark is valid both for inequality and equality constraints.

2.1. The Lagrangian Function

The introductory section of this chapter indicated that there is an important relationship between $\mathbf{g}^*$, the gradient of the cost function, and $\mathbf{a}_i^*$ $(i = 1, \ldots, m)$ the gradients of the constraint functions, all evaluated at a local minimizer. This has lead to the introduction of Lagrange's Function:

Definition 2.3. Lagrange's Function. Given the objective function f and the constraints c_i, $i = 1, \ldots, m$. Lagrange's function is defined by

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i c_i(\mathbf{x}) \, .$$

The scalars $\{\lambda_i\}$ are the *Lagrangian multipliers*

The gradient of L with respect to $\mathbf{x}$ is denoted $\mathbf{L}_x'$, and we see that

$$\mathbf{L}_x'(\mathbf{x}, \boldsymbol{\lambda}) \; = \; \mathbf{f}'(\mathbf{x}) - \sum_{i=1}^{n} \lambda_i \mathbf{c}_i'(\mathbf{x}) \, . \tag{2.4}$$

By comparison with the formulae in the introduction to this chapter we see that in all cases the necessary condition for a local, constrained minimizer could be expressed in the form $\mathbf{L}'_x(\mathbf{x}_s, \boldsymbol{\lambda}) = \mathbf{0}$.

For an unconstrained optimization problem you may recall that the necessary conditions and the sufficient condition for a minimizer involve the gradient $\mathbf{f}'(\mathbf{x}^*)$ and the Hessian matrix $\mathbf{f}''(\mathbf{x}^*)$ of the cost function, see Theorems 1.1, 1.2 and 1.5 in Frandsen et al (1999). In the next sections you will see that the corresponding results for constrained optimization will involve the gradient and the Hessian matrix (with respect to $\mathbf{x}$) of the Lagrangian function.

2.2. First Order Condition, Necessary Condition

First order conditions on local minimizers only consider first order partial derivatives of the cost function and the constraint functions. With this restriction we can only formulate the necessary conditions; the sufficient conditions also include second derivatives.

Our presentation follows Fletcher (1993), and we refer to this book for the formal proofs, which are not always straight forward. The strategy is as follows,

(1) Choose an arbitrary, feasible point.

(2) Determine a step which leads from this point to a neighbouring point, which is feasible and has a lower cost value.

(3) Detect circumstances which make the above impossible.

(4) Prove that only the above circumstances can lead to failure in step (2).

First, we formulate, the so-called first order *Karush–Kuhn–Tucker conditions* (*KKT conditions* for short):

Theorem 2.5. First order necessary conditions. (KKT conditions)

Assume that

a) $\mathbf{x}^*$ is a local constrained minimizer of f (see definition 1.4).

b) either b1) all active constraints c_i are linear,

or b2) the gradients $\mathbf{a}_i^* = \mathbf{c}_i'(\mathbf{x}^*)$ for all active constraints are linearly independent.

Then there exist Lagrangian multipliers $\{\lambda_i^*\}_{i=1}^m$ (see definition 2.3) such that

1° $\mathbf{L}'_x(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$,

2° $\lambda_i^* \geq 0,\ i = r+1, \ldots, m$,

3° $\lambda_i^* c_i(\mathbf{x}^*) = 0,\ i = 1, \ldots, m$.

The formulation is very compact, and we therefore give some clarifying remarks:

1° This was exemplified in connection with (2.4).

2° $\lambda_i^* \geq 0$ for all inequality constraints was exemplified in (2.2), and in Appendix A we give a formal proof.

3° For an equality constraint $c_i(\mathbf{x}^*) = 0$, and λ_i^* can have any sign.
For an active inequality constraint $c_i(\mathbf{x}^*) = 0$, and $\lambda_i^* \geq 0$.

For an inactive inequality constraint $c_i(\mathbf{x}^*) > 0$, so we must have $\lambda_i^* = 0$, confirming the observation in Example 1.4, that these constraints have no influence on the constrained minimizer.

In analogy with unconstrained optimization we can introduce the following

Corollary 2.6. Constrained stationary point

$\mathbf{x}_s$ is feasible and $(\mathbf{x}_s, \boldsymbol{\lambda}_s)$ satisfy 1°–3° in Theorem 2.5

$\Updownarrow$

$\mathbf{x}_s$ is a constrained stationary point

2.3. Second Order Conditions

The following example demonstrates that not only the curvature of the cost function but also the curvatures of the constraint functions are involved in the conditions for constrained minimizers.

Example 2.1. This example in $\mathbb{R}^2$ with one equality constraint $(r = m = 1)$ is due to Fiacco and McCormick (1968). The cost function and constraint are

$$f(\mathbf{x}) = \tfrac{1}{2}\left((x_1 - 1)^2 + x_2^2\right) , \qquad c_1(\mathbf{x}) = -x_1 + \beta x_2^2 .$$

We consider this problem for three different values of the parameter β, see Figure 2.7.

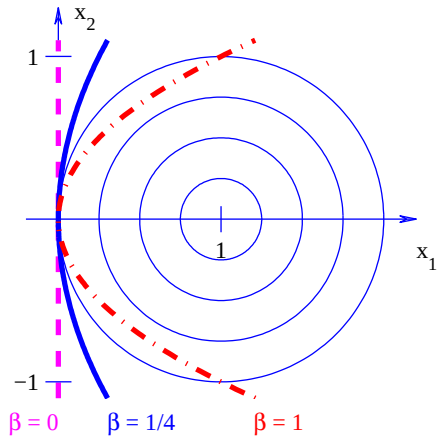


Figure 2.7: *Contours of f and the constraint*

$$-x_1 + \beta x_2^2 = 0$$

for three values of β.

In all the cases, $\mathbf{x}_s = \mathbf{0}$ is a constrained stationary point, see definition (2.6):

$$(\mathbf{x}, \lambda) = \tfrac{1}{2}\left((x_1 - 1)^2 + x_2^2\right) - \lambda\left(-x_1 + \beta x_2^2\right) ,$$

$$\mathbf{L}'_x(\mathbf{x}, \lambda) = \begin{bmatrix} x_1 - 1 \\ x_2 \end{bmatrix} - \lambda \begin{bmatrix} -1 \\ 2\beta x_2 \end{bmatrix}; \quad \mathbf{L}'_x(\mathbf{0}, \lambda) = \begin{bmatrix} -1 + \lambda \\ 0 \end{bmatrix} .$$

Thus, $(\mathbf{x}_s, \lambda_s) = (\mathbf{0},\ 1)$ satisfy 1° in Theorem 2.5, and 2°–3° are automatically satisfied when the problem has equality constraints only.

Notice, that f is *strictly convex* in $\mathbb{R}^2$.

For $\beta = 0$ the feasible region is the x_2-axis. This together with the contours of $f(\mathbf{x})$ near origo tells us that we have a local, constrained minimizer, $\mathbf{x}^* = \mathbf{0}$.

With $\beta = \frac{1}{4}$ the stationary point $\mathbf{x}_s = \mathbf{0}$ is also a local, constrained minimizer, $\mathbf{x}^* = \mathbf{0}$. This can be seen by correlating the feasible parabola with the contours of f around $\mathbf{0}$.

Finally, for $\beta = 1$ we get the rather surprizing result that $\mathbf{x}_s = \mathbf{0}$ is a local, constrained **maximizer**. Inspecting the feasible parabola and the contours carefully, you will discover that two local constrained minimizers have appeared around $\mathbf{x} = [0.5,\ \pm 0.7]^\top$. ∎

In Frandsen et al (2004) we derived the second order conditions for unconstrained minimizers. The derivation was based on the Taylor series (1.6) for $f(\mathbf{x}^* + \mathbf{h})$, and lead to conditions on the definiteness of the Hessian matrix $\mathbf{H}_u = \mathbf{f}''(\mathbf{x}_u)$, where $\mathbf{x}_u$ is the unconstrained minimizer.

The above example indicates that we have to take into account also the curvature of the active constraints, $\mathbf{A}_i^* = \mathbf{c}_i''(\mathbf{x}^*)$ for $i \in \mathcal{A}(\mathbf{x}^*)$.

The second order condition takes care of the situation where we move along the edge of $\mathcal{P}$ from a stationary point $\mathbf{x}$. Such a direction is called a *feasible active direction*:

Definition 2.7. Feasible active direction. Let $\mathbf{x} \in \mathcal{P}$. The nonzero vector $\mathbf{h}$ is a *feasible active direction* if

$$\mathbf{h}^\top \mathbf{c}_i'(\mathbf{x}) = 0$$

for all active constraints.

Now we use the Taylor series to study the variation of the Lagrangian function. Suppose we are at a constrained stationary point $\mathbf{x}_s$, and in the variation we keep $\boldsymbol{\lambda} = \boldsymbol{\lambda}_s$ from Definition 2.6. From $\mathbf{x}_s$ we move in a feasible active direction $\mathbf{h}$,

$$\begin{aligned} L(\mathbf{x}_s+\mathbf{h}, \boldsymbol{\lambda}_s) &= L(\mathbf{x}_s, \boldsymbol{\lambda}_s) + \mathbf{h}^\top \mathbf{L}'_x(\mathbf{x}_s, \boldsymbol{\lambda}_s) \\ &\quad + \tfrac{1}{2}\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{x}_s, \boldsymbol{\lambda}_s)\mathbf{h} + O(\|\mathbf{h}\|^3) \\ &= L(\mathbf{x}_s, \boldsymbol{\lambda}_s) + \tfrac{1}{2}\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{x}_s, \boldsymbol{\lambda}_s)\mathbf{h} + O(\|\mathbf{h}\|^3) \,, \end{aligned} \tag{2.8}$$

since $(\mathbf{x}_s, \boldsymbol{\lambda}_s)$ satisfies 1° in Theorem 2.5. The fact that $\mathbf{x}_s$ is a constrained stationary point implies that also 3° of Theorem 2.5 is satisfied, so that $L(\mathbf{x}_s, \boldsymbol{\lambda}_s) = f(\mathbf{x}_s)$. Since $\mathbf{h}$ is a feasible active direction, we obtain (again using 3° of Theorem 2.5),

$$\begin{aligned} L(\mathbf{x}_s+\mathbf{h}, \boldsymbol{\lambda}_s) &= f(\mathbf{x}_s+\mathbf{h}) - \textstyle\sum_{i=1}^m \lambda_i^{(s)} c_i(\mathbf{x}_s+\mathbf{h}) \\ &\simeq f(\mathbf{x}_s+\mathbf{h}) - \textstyle\sum_{i=1}^m \lambda_i^{(s)} (c_i(\mathbf{x}_s)+\mathbf{h}^\top \mathbf{c}'_i(\mathbf{x}_s)) \\ &= f(\mathbf{x}_s+\mathbf{h}) \,, \end{aligned} \tag{2.9}$$

and inserting this in (2.8) we get (for small values of $\|\mathbf{h}\|$)

$$f(\mathbf{x}_s+\mathbf{h}) \simeq f(\mathbf{x}_s) + \tfrac{1}{2}\mathbf{h}^\top \mathbf{W}_s \mathbf{h} \,, \tag{2.10a}$$

where the matrix $\mathbf{W}_s$ is given by

$$\mathbf{W}_s = \mathbf{L}''_{xx}(\mathbf{x}_s, \boldsymbol{\lambda}_s) = \mathbf{f}''(\mathbf{x}_s) - \sum_{i=1}^m \lambda_i^{(s)} \mathbf{c}''_i(\mathbf{x}_s) \,. \tag{2.10b}$$

This leads to the sufficient condition that the stationary point $\mathbf{x}_s$ is a local, constrained minimizer if $\mathbf{h}^\top \mathbf{W}_s \mathbf{h} > 0$ for any feasible active direction $\mathbf{h}$. Since this condition is also necessary we can formulate the following two second order conditions:

Theorem 2.11. Second order necessary condition.
Assume that

a) $\mathbf{x}^*$ is a local constrained minimizer for f.
b) As b) in Theorem 2.5.
c) All the active constraints are *strongly active*.

Then there exists Lagrangian multipliers $\{\lambda_i^*\}_{i=1}^m$ (see Definition 2.3) such that

1° $\mathbf{L}'_x(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$,

2° $\lambda_i^* \geq 0,\ i = r+1, \ldots, m$,

3° $\lambda_i^* > 0$ if c_i is active, $i = r+1, \ldots, m$,

4° $\lambda_i^* c_i(\mathbf{x}^*) = 0,\ i = 1, \ldots, m$,

5° $\mathbf{h}^\top \mathbf{W}^* \mathbf{h} \geq 0$ for any feasible active direction $\mathbf{h}$.
Here, $\mathbf{W}^* = \mathbf{L}''_{xx}(\mathbf{x}^*, \boldsymbol{\lambda}^*)$.

Theorem 2.12. Second order sufficient condition.
Assume that

a) $\mathbf{x}_s$ is a local constrained stationary point (see Definition 2.6).
b) As b) in Theorem 2.5.
c) As c) in Theorem 2.11.
d) $\mathbf{h}^\top \mathbf{W}^* \mathbf{h} > 0$ for any feasible active direction $\mathbf{h}$,
where $\mathbf{W}^* = \mathbf{L}''_{xx}(\mathbf{x}^*, \boldsymbol{\lambda}^*)$.

Then
$\mathbf{x}_s$ is a local constrained minimizer.

For the proofs we refer to Fletcher (1993). There, you may also find a treatment of the cases, where the gradients of the active constraints are not linearly independent, and where some constraints are weakly active.

Example 2.2. Continuing from Example 2.1 we find

$$\mathbf{L}''_{xx}(\mathbf{x},\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 0 & 0 \\ 0 & 2\beta \end{bmatrix} .$$

At the stationary point $\mathbf{x}_s = \mathbf{0}$ we found $\lambda_s = 1$. Further, from Figure 2.7 and Definition 2.7 we see that $\mathbf{h} = [0\ h_2]^\top$ is the only feasible active direction, and we get

$$\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{h}_s,\lambda_s)\mathbf{h} = (1-2\beta)h_2^2 .$$

This is positive if $\beta < \frac{1}{2}$, and Theorem 2.12 shows that in this case $\mathbf{x}_s = \mathbf{0}$ is a local, constrained minimizer.

If $\beta > \frac{1}{2}$, then $\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{h}_s,\lambda_s)\mathbf{h} < 0$, contradicting 5° in Theorem 2.11; therefore $\mathbf{x}_s = \mathbf{0}$ cannot be a local, constrained minimizer when $\beta > \frac{1}{2}$.

The limiting case $\beta = \frac{1}{2}$ is not covered by the theorems. In order to investigate it, higher order terms are needed in the Taylor expansion for $L(\mathbf{x}_s+\mathbf{h},\boldsymbol{\lambda}_s)$. ∎

Finally, we give the following theorem, whose proof can be found p 10 in Madsen (1995):

Theorem 2.13. Third sufficient condition. Assume that
a) $\mathbf{x}_s$ is a local constrained stationary point (see Definition 2.6),
b) all active constraints are linear,
c) $\mathbf{h}^\top \mathbf{W}^* \mathbf{h} > 0$ for any feasible active direction $\mathbf{h} \neq \mathbf{0}$.

Then
$\mathbf{x}_s$ is a local constrained minimizer.

3. QUADRATIC OPTIMIZATION

We now start to introduce solution methods for different classes of optimization problems with constraints. The fundamental class has linear cost functions and also linear constraints. This class is called *linear optimization problems* and is covered in Nielsen (1999).

The next class has a quadratic cost function and all the constraints are linear. We call it

Definition 3.1. The quadratic optimization problem (QO).
Find

$$\mathbf{x}^* = \text{argmin}_{\mathbf{x}\in\mathcal{P}}\{q(\mathbf{x})\} ,$$

where

$$q(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^\top \mathbf{H}\,\mathbf{x} + \mathbf{g}^\top \mathbf{x} ,$$

$$\mathcal{P} = \{\mathbf{x}\in\mathbf{R}^n \mid \mathbf{a}_i^\top \mathbf{x} = b_i ,\ \ i=1,\ldots,r$$
$$\mathbf{a}_i^\top \mathbf{x} \geq b_i ,\ \ i=r+1,\ldots,m\ \} .$$

The matrix $\mathbf{H}\in\mathbf{R}^{n\times n}$ and the vectors $\mathbf{g},\mathbf{a}_1,\ldots,\mathbf{a}_m\in\mathbf{R}^n$ are given. The associated Lagrange function is

$$L(\mathbf{x},\boldsymbol{\lambda}) = \tfrac{1}{2}\mathbf{x}^\top \mathbf{H}\,\mathbf{x} + \mathbf{g}^\top \mathbf{x} - \sum_{i=1}^{m} \lambda_i(\mathbf{a}_i^\top \mathbf{x} - b_i) , \tag{3.2a}$$

with the first and second order derivatives

$$\mathbf{L}'_x(\mathbf{x},\boldsymbol{\lambda}) = \mathbf{H}\mathbf{x} + \mathbf{g} - \sum_{i=1}^{m} \lambda_i \mathbf{a}_i, \qquad \mathbf{L}''_{xx}(\mathbf{x},\boldsymbol{\lambda}) = \mathbf{H} . \tag{3.2b}$$

Throughout this chapter we have the assumptions

Assumption 3.3. $\mathbf{H}$ is symmetric and positive definite.

(See Fletcher (1993) for methods for the cases where these simplifying assumptions are not satisfied). Under Assumption 3.3 the problem is strictly convex (Theorem 1.16). This ensures that $q(\mathbf{x})\to+\infty$ when $\|\mathbf{x}\|\to\infty$, irrespective of the direction. Thus we need not require the feasible region $\mathcal{P}$ to be bounded. All the constraint functions are linear and this makes $\mathcal{P}$ convex. Thus, in this case Theorem 1.17 leads to

Corollary 3.4. Under Assumption 3.3 the problem QO of Definition 3.1 has a unique solution.

As in Chapter 2 we shall progress gradually with the different complications of the methods, ending the chapter with a method for non-linear optimization using iterations where each step solves a quadratic optimization problem, gradually approaching the properties of the non-linear cost function and constraints.

Example 3.1. In Figure 3.1 you see the contours of a positive definite quadratic in $\mathbb{R}^2$. If there are no constraints on the minimizer, we get the unconstrained minimizer, indicated by $\mathbf{x}_\mathrm{u}$ in the figure.

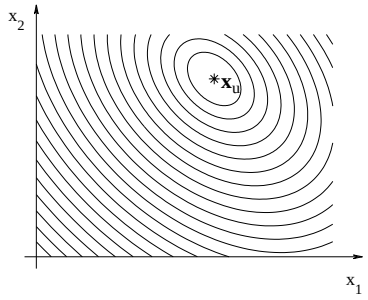


Figure 3.1: *Contours of a quadratic in* $\mathbb{R}^2$ *and its unconstrained minimizer* $\mathbf{x}_\mathrm{u}$

■

The solution of the unconstrained quadratic optimization problem corresponding to Definition 3.1 is found from the necessary condition $\mathbf{q}'(\mathbf{x}_\mathrm{u})=\mathbf{0}$

which is the following linear system of equations,

$$\mathbf{H}\,\mathbf{x}_\mathrm{u} = -\mathbf{g}\,. \tag{3.5}$$

The solution is unique according to our assumptions.

3.1. Basic Quadratic Optimization

The basic quadratic optimization problem is the special case of Problem QO (Definition 3.1) with only equality constraints, ie $m=r$. We state it in the form[1)]

Definition 3.6. Basic quadratic optimization problem (BQO)
Find

$$\mathbf{x}^* \;=\; \mathrm{argmin}_{\mathbf{x}\in\mathcal{P}}\{q(\mathbf{x})\}\,,$$

where

$$q(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^\top\mathbf{H}\,\mathbf{x}+\mathbf{g}^\top\mathbf{x}\,, \qquad \mathcal{P}=\{\mathbf{x}\in\mathbb{R}^n \mid \mathbf{A}^\top\mathbf{x}=\mathbf{b}\}\,.$$

The matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$ has the columns $\mathbf{A}_{:,j}=\mathbf{a}_j$ and b_j is the jth element in $\mathbf{b}\in\mathbb{R}^m$.

The solution can be found directly, namely by solving the linear system of equations which express the necessary condition that the Lagrange function L is stationary at the solution with respect to both of its vector variables $\mathbf{x}$ and $\boldsymbol{\lambda}$:

$$\begin{aligned} \mathbf{L}'_x(\mathbf{x},\boldsymbol{\lambda})=\mathbf{0}: &\qquad \mathbf{H}\mathbf{x}+\mathbf{g}-\mathbf{A}\boldsymbol{\lambda}=\mathbf{0}\,,\\ \mathbf{L}'_\lambda(\mathbf{x},\boldsymbol{\lambda})=\mathbf{0}: &\qquad \mathbf{A}^\top\mathbf{x}-\mathbf{b}=\mathbf{0}\;. \end{aligned} \tag{3.7}$$

The first equation is the KKT condition, and the second expresses that the constraints are satisfied at the solution. This linear system of equations has the dimension $(n+r)\times(n+r)$, with $r=m$. Thus the solution requires $O((n+m)^3)$ operations. We return to the solution of (3.7) in Section 3.3.

1) In other presentations you may find the constraint equation formulated as $\tilde{\mathbf{A}}\mathbf{x}=\mathbf{b}$ with $\tilde{\mathbf{A}}=\mathbf{A}^\top$. Hopefully this will not lead to confusion.

3.2. General Quadratic Optimization

In the general case we have both equality constraints and inequality constraints in Problem 3.1, and we must use an iterative method to solve the problem. If we knew which constraints are active at the solution $\mathbf{x}^*$ we could set up a linear system like (3.7) and find the solution directly. Thus the problem can be formulated as that of finding the active set $\mathcal{A}(\mathbf{x}^*)$.

We present a so-called *active set method*. Each iterate $\mathbf{x}$ is found via an active set $\mathcal{A}$ (corresponding to the constraints that should be satisfied with "=", cf Definition 1.3). Ignoring the inactive constraints we consider the basic quadratic optimization problem with the equality constraints given by $\mathcal{A}$:

Definition 3.8. Current BQO problem (CBQO($\mathcal{A}$))
Find

$$\mathbf{x}_{\text{eq}} = \text{argmin}_{\mathbf{x}\in\mathcal{P}}\{q(\mathbf{x})\} ,$$

where

$$q(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^\top \mathbf{H}\,\mathbf{x} + \mathbf{g}^\top \mathbf{x} , \qquad \mathcal{P} = \{\mathbf{x}\in\mathbb{R}^n \mid \mathbf{A}^\top\mathbf{x} = \mathbf{b}\} .$$

The matrix $\mathbf{A}\in\mathbb{R}^{n\times p}$ has the columns $\mathbf{A}:_{,j} = \mathbf{a}_j$, $j\in\mathcal{A}$ and $\mathbf{b}\in\mathbb{R}^p$ has the corresponding values of b_j. p is the number of elements in $\mathcal{A}$.

We shall refer to *CBQO($\mathcal{A}$)* as a function (subprogram) that returns $(\mathbf{x}_{\text{eq}}, \boldsymbol{\lambda}_{\text{eq}})$, the minimizer and the corresponding set of Lagrange multipliers corresponding to the active set $\mathcal{A}$. Similar to the BQO they are found as the solution to the following linear system of dimension $(n+p)\times(n+p)$:

$$\begin{aligned} &\mathbf{H}\mathbf{x} + \mathbf{g} - \mathbf{A}\boldsymbol{\lambda} = \mathbf{0} , \\ &\mathbf{A}^\top\mathbf{x} - \mathbf{b} = \mathbf{0} \ . \end{aligned} \tag{3.9}$$

In the iteration for solving Problem 3.1 all iterates are feasible. This means that we have a feasible $\mathbf{x}$ and an active set $\mathcal{A}$ at the beginning of each iteration. Now the CBQO (Definition 3.8) is solved. If $\mathbf{x}_{\text{eq}}$ violates some constraint (ie some of the ignored inequality constraints), then the next iterate is that feasible point on the line from $\mathbf{x}$ to $\mathbf{x}_{\text{eq}}$ which is closest to $\mathbf{x}_{\text{eq}}$, and the new inequality constraint(s) becoming active is added to $\mathcal{A}$.

If, on the other hand, $\mathbf{x}_{\text{eq}}$ is feasible, then we are finished (ie $\mathbf{x}^* = \mathbf{x}_{\text{eq}}$) provided that all Lagrange multipliers corresponding to $\widetilde{\mathcal{A}}$ are non-negative (Theorem 2.13). If there is one or more $\lambda_j < 0$ for $j\in\widetilde{\mathcal{A}}$ (ie for active inequality constraints), then one of the corresponding indices is dropped from $\mathcal{A}$ before the next iteration.

Before formally defining the strategy in Algorithm 3.10 we illustrate it through a simple example.

Example 3.2. We take a geometric view of a problem in $\mathbb{R}^2$ with 3 inequality constraints. In Figure 3.2 we give the contours of the cost function and the border lines for the inequalities. The infeasible side is hatched.

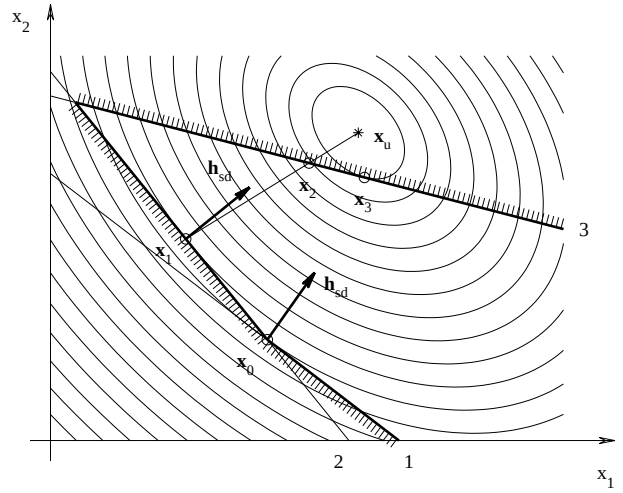


Figure 3.2: *Contours of a quadratic optimization problem in* $\mathbb{R}^2$ *with 3 inequality constraints,* $\mathbf{a}_i^\top\mathbf{x} \geq b_i,\ i = 1, 2, 3$

The starting point $\mathbf{x} = \mathbf{x}_0$ is feasible, and we define $\mathcal{A} \equiv \mathcal{A}(\mathbf{x}_0) = \{1, 2\}$, while the third constraint is inactive. The "pulling force" $\mathbf{h}_{\text{sd}}$ $(=-\mathbf{q}'(\mathbf{x}_0))$ shows that we should leave inequality no. 1. This corresponds to the fact that $\lambda_1 < 0$. Thus the next active set is $\mathcal{A} = \{2\}$. The solution to the the corresponding system (2.13) is $\mathbf{x} = \mathbf{x}_1$. This is feasible, but the "pulling force" tells us that we should

loosen the only remaining constraint (corresponding to $\lambda_2 < 0$). Thus, the next CBQO step will lead to $\mathbf{x}_{\text{u}}$, the unconstrained minimizer which is infeasible: It satisfies constraints 1 and 2, but not 3. The next iterate, $\mathbf{x} = \mathbf{x}_2$ is found as the intersection between the line from $\mathbf{x}_1$ to $\mathbf{x}_{\text{u}}$ and the bordering line for $\mathbf{a}_3^\top \mathbf{x} \geq b_3$.

Finally, a CBQO step from $\mathbf{x}_2$ with $\mathcal{A} = \{3\}$ gives $\mathbf{x} = \mathbf{x}_3$. This is feasible and by checking the contours of the cost function we see that we have come to the solution, $\mathbf{x}^* = \mathbf{x}_3$. Algebraically we see this from the fact that $\lambda_3 > 0$. ∎

The strategy from this example is generalized in Algorithm 3.10.

Algorithm 3.10. General quadratic optimization

begin
  $\mathbf{x} := \mathbf{x}_0$ $\{1°\}$
  $\mathcal{A} := \mathcal{A}(\mathbf{x})$ $\{2°\}$
  stop := **false**
  repeat
    $(\mathbf{x}_{\text{eq}}, \boldsymbol{\lambda}_{\text{eq}}) :=$ *CBQO(*$\mathcal{A}$*)* cf Definition 3.8 and (3.9)
    if $\mathbf{x}_{\text{eq}}$ is infeasible
      $\mathbf{x} :=$ best feasible point on the line from $\mathbf{x}$ to $\mathbf{x}_{\text{eq}}$ $\{3°\}$
      Update $\mathcal{A}$ $\{3°\}$
    else
      $\mathbf{x} := \mathbf{x}_{\text{eq}}$
      $\mathcal{L} := \{j \in \widetilde{\mathcal{A}} \mid \lambda_j < 0\}$
      if $\mathcal{L}$ is empty
        stop := **true** $\{4°\}$
      else
        Remove an element of $\mathcal{L}$ from $\mathcal{A}$ $\{5°\}$
  until stop
end

We have the following remarks:

1° The initial point $\mathbf{x}_0$ must be feasible. How to find such a point is discussed in Section 3.3.

2° $\mathcal{A}$ holds the indices of current active constraints, cf Definition 1.3.

3° The vector $\mathbf{x}_{\text{eq}}$ satisfies the current active constraints, but some of the inequality constraints that were ignored: $j \in \{r+1, \ldots, m\} \setminus \widetilde{\mathcal{A}}$ may be violated at $\mathbf{x}_{\text{eq}}$. Let $\mathcal{V}$ denote the set of indices of constraints violated at $\mathbf{x}_{\text{eq}}$, $\mathcal{V} = \{j \in \{r+1, \ldots, m\} \mid \mathbf{a}_j^\top \mathbf{x}_{\text{eq}} < b_j\}$. We shall choose the best feasible point on the line between $\mathbf{x}$ and $\mathbf{x}_{\text{eq}}$,

$$\widetilde{\mathbf{x}} = \mathbf{x} + t(\mathbf{x}_{\text{eq}} - \mathbf{x}), \quad 0 < t < 1 \,. \tag{3.11a}$$

The value of t which makes constraint no. j active is given by $\mathbf{a_j}^\top \widetilde{\mathbf{x}} = b_j$, which is equivalent to

$$t_j = (b_j - \mathbf{a_j}^\top \mathbf{x}) / \mathbf{a_j}^\top (\mathbf{x}_{\text{eq}} - \mathbf{x}) \,. \tag{3.11b}$$

Since $\mathbf{x}$ is feasible and $\mathbf{x}_{\text{eq}}$ is optimal in CBQO, and since the objective function is convex, the best feasible point on the line is the feasible point closest to $\mathbf{x}_{\text{eq}}$. This corresponds to

$$k = \operatorname{argmin}_{j \in \mathcal{V}} t_j \quad , \tag{3.11c}$$

The new $\mathbf{x}$ is found by (3.11a) with $t = t_k$, and the index k is added to $\mathcal{A}$. If the minimum in (3.11c) is taken by several values of k then all of these are added to $\mathcal{A}$.

4° Since $\mathbf{x}_{\text{eq}}$ is feasible and Lagrange multipliers corresponding to inequality constraints are nonnegative, $\mathbf{x}_{\text{eq}} = \mathbf{x}^*$ solves the problem according to Theorem 2.13.

5° If an active inequality constraint at the CBQO solution has a negative Lagrange multiplier, then we can reduce the cost function by loosening this constraint.

Finite termination. For each choice $\mathcal{A}$ of currently active constraints CBQO has a unique minimizer $\mathbf{x}_{\text{eq}}$. Each time an element of $\mathcal{L}$ is removed from $\mathcal{A}$ (see remark 5°) we have $\mathbf{x} = \mathbf{x}_{\text{eq}}$ and there is a strict decrease in the objective function: $q(\mathbf{x}_{\text{new}}) < q(\mathbf{x})$. Since each new iterate satisfies

$q(\mathbf{x}_{\text{new}}) \le q(\mathbf{x})$ the strict decrease can only take place a finite number of times because of the finite number of possible active sets $\mathcal{A}$.

Therefore we only drop a constraint a finite number of times, and thus cycling cannot take place: The algorithm must stop after a finite number of iterations.

3.3. Implementation Aspects

To start Algorithm 3.10 we need a feasible starting point $\mathbf{x}_0$. This is simple if $m \le n$ (the number of constraints is at most equal to the number of unknown): We just solve

$$\mathbf{A}^\top \mathbf{x} = \mathbf{b} \,, \tag{3.12a}$$

with $\mathbf{A} \in \mathbf{R}^{n\times m}$ having the columns $\mathbf{a}_i,\ i = 1, \ldots, m$. If $m < n$, then this system is underdetermined, and the solution has (at least) $n-m$ free parameters. For any choice of these the vector $\mathbf{x}$ is feasible; all the constraints are active.

If $m > n$, we cannot expect to find an $\mathbf{x}$ with all inequality constraints active. Instead, we can use the formulation

$$\mathbf{A}^\top \mathbf{x} - \mathbf{s} = \mathbf{b} \quad \text{with } \mathbf{s} \ge \mathbf{0} \,, \tag{3.12b}$$

and $s_i = 0$ for the equality constraints. The problem of finding an $\mathbf{x}$ that satisfies (3.12b) is similar to getting a feasible starting point for the SIMPLEX method in Linear Optimization, see Section 4.4 in Nielsen (1999).

The most expensive part of the process is solution of the CBQO at each iteration. The simplest approach would be to start from scratch for each new $\mathcal{A}$. Then the accumulated cost of the computations involved in the solutions of (3.9) would be $O((n+m)^3)$ floating point operations per call of CBQO. If constraint gradients are linearly independent then the number of equality and active constraints cannot exceed n, and thus the work load is $O(n^3)$ floating point operations per call of CBQO.

Considerable savings are possible when we note that each new $\mathbf{A}$ is obtained from the previous either by deleting a column or by adding one or more new

columns. First, we note that the matrix $\mathbf{H}$ is used in all iteration steps. It should be factorized once and for all, eg by Cholesky's method, cf Appendix A in Frandsen et al (2004),

$$\mathbf{H} \;=\; \mathbf{C}\mathbf{C}^\top \,,$$

where $\mathbf{C}$ is lower triangular. This requires $O(n^3)$ operations, and after this, each "$\mathbf{H}^{-1}\mathbf{w}$" will then require $O(n^2)$ operations.

The first equation in (3.9) can be reformulated to

$$\mathbf{x} \;=\; \mathbf{H}^{-1}(\mathbf{A}\boldsymbol{\lambda} - \mathbf{g}) \,, \tag{3.13a}$$

and when we insert this in the second equation in (3.9), we get

$$(\mathbf{A}^\top \mathbf{H}^{-1} \mathbf{A})\boldsymbol{\lambda} \;=\; \mathbf{b} + \mathbf{A}^\top \mathbf{H}^{-1} \mathbf{g} \,. \tag{3.13b}$$

Next, we can reformulate (3.13b) to

$$\mathbf{G}\boldsymbol{\lambda} = \mathbf{b} + \mathbf{A}^\top \mathbf{d}$$
$$\text{with } \mathbf{G} = (\mathbf{C}^{-1}\mathbf{A})^\top (\mathbf{C}^{-1}\mathbf{A}) \,,\ \mathbf{d} = \mathbf{H}^{-1}\mathbf{g} \,.$$

This system is solved via the Cholesky factorization of the $p \times p$ matrix $\mathbf{G}$ (p being the current number of active constraints). When $\mathbf{A}$ changes by adding or deleting a column, it is possible to update this factorization in $O(n \cdot p)$ operations, and the cost of each iteration step reduces to $O(n^2)$ operations. For more details see pp 18–19 in Madsen (1995).

There are alternative methods for solving the system (3.9). Gill and Murray (1974) suggest to use the *QR factorization*[2] of the active constraint matrix,

$$\mathbf{A} \;=\; \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \;=\; \begin{bmatrix} \mathbf{Q}_{\text{R}} & \mathbf{Q}_{\text{N}} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \;=\; \mathbf{Q}_{\text{R}} \mathbf{R} \,, \tag{3.14}$$

where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular. As indicated, we can split $\mathbf{Q}$ into $\mathbf{Q}_{\text{R}} \in \mathbf{R}^{n\times p}$ and $\mathbf{Q}_{\text{N}} \in \mathbf{R}^{n\times(n-p)}$. The orthogonality of $\mathbf{Q}$ implies that

[2] See eg Chapter 2 in Madsen and Nielsen (2002) or Section 5.2 in Golub and Van Loan (1996).

$$\mathbf{Q}_{\mathrm{R}}^{\top}\mathbf{Q}_{\mathrm{R}} = \mathbf{I}_{(n-p)\times(n-p)}\ , \qquad \mathbf{Q}_{\mathrm{R}}^{\top}\mathbf{Q}_{\mathrm{N}} = \mathbf{0}_{(n-p)\times p}\ , \tag{3.15}$$

where the indices on $\mathbf{I}$ and $\mathbf{0}$ are the dimensions of the matrix. The columns of $\mathbf{Q}$ form an orthonormal basis of $\mathbb{R}^n$, and we can express $\mathbf{x}$ in the form

$$\mathbf{x} = \mathbf{Q}_{\mathrm{R}}\mathbf{u} + \mathbf{Q}_{\mathrm{N}}\mathbf{v}, \qquad \mathbf{u}\in\mathbb{R}^p,\ \ \mathbf{v}\in\mathbb{R}^{n-p}\ . \tag{3.16}$$

Inserting (3.14) – (3.16) in the second equation of (3.9) we get

$$\mathbf{R}^{\top}\mathbf{Q}_{\mathrm{R}}^{\top}(\mathbf{Q}_{\mathrm{R}}\mathbf{u} + \mathbf{Q}_{\mathrm{N}}\mathbf{v}) \ =\ \mathbf{R}^{\top}\mathbf{u} \ =\ \mathbf{b}\ .$$

This lower triangular system is solved by forward substitution. To find $\mathbf{v}$ in (3.16) we multiply the first equation of (3.9) by $\mathbf{Q}_{\mathrm{N}}^{\top}$ and get

$$\mathbf{Q}_{\mathrm{N}}^{\top}\mathbf{H}(\mathbf{Q}_{\mathrm{R}}\mathbf{u} + \mathbf{Q}_{\mathrm{N}}\mathbf{v}) + \mathbf{Q}_{\mathrm{N}}^{\top}\mathbf{g} - \mathbf{Q}_{\mathrm{N}}^{\top}\mathbf{Q}_{\mathrm{R}}\mathbf{R}\boldsymbol{\lambda} \ =\ \mathbf{0}\ ,$$

and by use of the second identity in (3.15) this leads to

$$\mathbf{Q}_{\mathrm{N}}^{\top}\mathbf{H}\mathbf{Q}_{\mathrm{N}}\,\mathbf{v} \ =\ -\mathbf{Q}_{\mathrm{N}}^{\top}(\mathbf{H}\mathbf{Q}_{\mathrm{R}}\mathbf{u} + \mathbf{g})\ . \tag{3.17}$$

The $(n-p)\times(n-p)$ matrix $\mathbf{M} = \mathbf{Q}_{\mathrm{N}}^{\top}\mathbf{H}\mathbf{Q}_{\mathrm{N}}$ is symmetric and positive definite, and (3.17) can be solved via Cholesky factorization of $\mathbf{M}$. Finally, $\boldsymbol{\lambda}$ can be computed from the first equation of (3.9):

$$\mathbf{Q}_{\mathrm{R}}^{\top}\mathbf{A}\boldsymbol{\lambda} \ =\ \mathbf{R}\boldsymbol{\lambda} \ =\ \mathbf{Q}_{\mathrm{R}}^{\top}(\mathbf{H}\mathbf{x} + \mathbf{g})\ . \tag{3.18}$$

We used (3.14) and (3.15) in the first reformulation, and $\mathbf{x}$ is given by (3.16). The system (3.18) is solved by back substitution.

There are efficient methods for updating the QR factorization of $\mathbf{A}$, when this matrix is changed because an index is added to or removed from the active set, see eg Section 12.5 in Golub and van Loan (1996). This method for solving the system (3.9) is advantageous if p is large, $p \gtrsim \frac{1}{2}\,n$.

If the problem is large and *sparse*, ie most of the elements in $\mathbf{H}$ and $\mathbf{A}$ are zero, then both the above approaches tend to give matrices that are considerably less sparse. In such cases it is recommended to solve (3.9) via the so-called *augmented system*,

$$\begin{bmatrix} \mathbf{H} & -\mathbf{A} \\ -\mathbf{A}^{\top} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} \ =\ -\begin{bmatrix} \mathbf{g} \\ \mathbf{b} \end{bmatrix}\ . \tag{3.19}$$

Because of Assumption 3.3 the matrix is symmetric. It is not positive definite, however[3], but there are efficient methods for solving such systems, where the sparsity is preserved better, without spoiling numerical stability. It is also possible to handle the updating aspects efficiently; see eg Duff (1993).

3.4. Sequential Quadratic Optimization

A number of efficient methods for *non-linear optimization* originate from *sequential quadratic optimization*. These methods are iterative methods where each iteration step includes the solution of a general quadratic optimization problem.

First, we consider problems with equality constraints, only:

$$\begin{aligned} &\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{P}} f(\mathbf{x})\ , \\ &\mathcal{P} = \{\mathbf{x}\in\mathbb{R}^n \mid \mathbf{c}(\mathbf{x}) = \mathbf{0}\}\ . \end{aligned} \tag{3.20}$$

Here, $\mathbf{c}$ is the vector function $\mathbf{c} : \mathbb{R}^n \mapsto \mathbb{R}^r$, whose ith component is the ith constraint function c_i.

The corresponding Lagrange's function (Definition 2.3) is

$$L(\mathbf{x},\boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^{\top}\mathbf{c}(\mathbf{x})\ , \tag{3.21a}$$

with the gradient

$$\mathbf{L}'(\mathbf{x},\boldsymbol{\lambda}) = \begin{bmatrix} \mathbf{L}_x'(\mathbf{x},\boldsymbol{\lambda}) \\ \mathbf{L}_{\lambda}'(\mathbf{x},\boldsymbol{\lambda}) \end{bmatrix} \ =\ \begin{bmatrix} \mathbf{f}'(\mathbf{x}) - \mathbf{J}_{\mathbf{c}}{}^{\top}\boldsymbol{\lambda} \\ -\mathbf{c}(\mathbf{x}) \end{bmatrix}\ , \tag{3.21b}$$

where $\mathbf{J}_c$ is the *Jacobian matrix* of $\mathbf{c}$,

$$(\mathbf{J}_c)_{ij} = \frac{\partial c_i}{\partial x_j}(\mathbf{x}) \quad\Longleftrightarrow\quad \mathbf{J}_c = [\mathbf{c}_1'(\mathbf{x})\ \cdots\ \mathbf{c}_r'(\mathbf{x})]^{\top}\ . \tag{3.21c}$$

At a stationary point $\mathbf{x}_{\mathrm{s}}$ with corresponding $\boldsymbol{\lambda}_{\mathrm{s}}$ we have $\mathbf{L}'(\mathbf{x}_{\mathrm{s}},\boldsymbol{\lambda}_{\mathrm{s}}) = \mathbf{0}$, which includes that $\mathbf{c}(\mathbf{x}) = \mathbf{0}$ (the constraints are satisfied) and

[3] A necessary condition for a symmetric matrix to be positive definite is that all the diagonal elements are strictly positive.

$$\mathbf{f}'(\mathbf{x}) - \mathbf{J_c}^\top \boldsymbol{\lambda} = \mathbf{0} \,,$$

which we recognize as a part of the KKT conditions (Theorem 2.5).

Thus we can reformulate problem (3.20) to a non-linear system of equations: Find $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ such that

$$\mathbf{L}'(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \,.$$

We can use Newton-Raphson's method to solve this problem. In each iteration step with current iterate $(\mathbf{x}, \boldsymbol{\lambda})$, we find the next iterate as $(\mathbf{x}+\mathbf{h}, \boldsymbol{\lambda}+\boldsymbol{\eta})$, with the step determined by

$$\mathbf{L}''(\mathbf{x}, \boldsymbol{\lambda}) \begin{bmatrix} \mathbf{h} \\ \boldsymbol{\eta} \end{bmatrix} \;=\; -\mathbf{L}'(\mathbf{x}, \boldsymbol{\lambda}) \,,$$

where $\mathbf{L}''$ is the total Hessian,

$$\mathbf{L}'' = \begin{bmatrix} \mathbf{L}''_{xx} & \mathbf{L}''_{x\lambda} \\ \mathbf{L}''_{\lambda x} & \mathbf{L}''_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{W} & -\mathbf{J_c}^\top \\ -\mathbf{J}_c & \mathbf{0} \end{bmatrix}$$

with

$$\mathbf{W} = \mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{f}''(\mathbf{x}) - \textstyle\sum_{i=1}^{r} \lambda_i \mathbf{c}_i''(\mathbf{x}) \,.$$

One Newton-Raphson step is

$$\begin{bmatrix} \mathbf{W} & -\mathbf{J_c}^\top \\ -\mathbf{J}_c & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \boldsymbol{\eta} \end{bmatrix} \;=\; -\begin{bmatrix} \mathbf{f}'(\mathbf{x}) - \mathbf{J_c}^\top \boldsymbol{\lambda} \\ -\mathbf{c}(\mathbf{x}) \end{bmatrix} \,,$$
$$\mathbf{x} := \mathbf{x} + \mathbf{h}; \quad \boldsymbol{\lambda} := \boldsymbol{\lambda} + \boldsymbol{\eta} \,,$$

an by elimination of $\boldsymbol{\eta}$ we obtain

$$\begin{bmatrix} \mathbf{W} & -\mathbf{J_c}^\top \\ -\mathbf{J}_c & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \boldsymbol{\lambda} \end{bmatrix} = -\begin{bmatrix} \mathbf{f}'(\mathbf{x}) \\ -\mathbf{c}(\mathbf{x}) \end{bmatrix} \,, \qquad (3.22)$$
$$\mathbf{x} := \mathbf{x} + \mathbf{h} \,.$$

What has this got to do with Quadratic Optimization? Quite a lot! Compare (3.22) with (3.19). Since (3.19) gives the solution $(\mathbf{x}, \boldsymbol{\lambda})$ to the CBQO, (Definition 3.8), it follows that (3.22) gives the solution $\mathbf{h}$ and the corresponding Lagrange multiplier vector $\boldsymbol{\lambda}$ to the following problem,

$$\text{Find } \mathbf{h} \;=\; \text{argmin}_{\mathbf{h}\in\mathcal{P}_{\text{lin}}}\{q(\mathbf{h})\} \qquad (3.23a)$$

where

$$\begin{aligned} q(\mathbf{h}) &= \tfrac{1}{2}\mathbf{h}^\top \mathbf{W}\, \mathbf{h} + \mathbf{f}'(\mathbf{x})^\top \mathbf{h} \\ \mathcal{P}_{\text{lin}} \;&= \{\mathbf{h} \in \mathbb{R}^n \mid \mathbf{J}_c\mathbf{h} + \mathbf{c}(\mathbf{x}) = \mathbf{0}\} \end{aligned} \qquad (3.23b)$$

Adding a constant to q makes no difference to the solution vector. If we furthermore insert the value of $\mathbf{W}$ then (3.23b) becomes

$$\begin{aligned} q(\mathbf{h}) &= \tfrac{1}{2}\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda})\, \mathbf{h} + \mathbf{f}'(\mathbf{x})^\top \mathbf{h} + f(\mathbf{x}) \\ \mathcal{P}_{\text{lin}} \;&= \{\mathbf{h} \in \mathbb{R}^n \mid \mathbf{J}_c\mathbf{h} + \mathbf{c}(\mathbf{x}) = \mathbf{0}\} \,. \end{aligned} \qquad (3.23c)$$

By comparison with the Taylor expansions (1.6) and (1.7) we see that if $\boldsymbol{\lambda} = \mathbf{0}$ then $q(\mathbf{h})$ is a second order approximation to $f(\mathbf{x}+\mathbf{h})$, and $\mathbf{J}_c\mathbf{h} + \mathbf{c}(\mathbf{x})$ is a first order approximation to $\mathbf{c}(\mathbf{x}+\mathbf{h})$. In other words, (3.23) represents a local QO (ie Quadratic Optimization) approximation to (3.20), except for the fact that $\mathbf{f}''(\mathbf{x})$ is replaced by $\mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda})$. However, using $\mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda})$ provides faster convergence than using a quadratic approximation to f, which follows from this argument: It is shown above that solving (3.23) and subsequently letting

$$\mathbf{x} := \mathbf{x} + \mathbf{h}$$

in the final stages of an iterative method for solving (3.20) corresponds to applying the Newton-Raphson method to find a stationary point of the Lagrange function L. Under the usual regularity assumptions this provides quadratic final convergence to the solution of (3.20). Using $\mathbf{f}''(\mathbf{x})$ instead of $\mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda})$ in the QO approximation would perturb the Newton-Raphson matrix (except for $\boldsymbol{\lambda} = \mathbf{0}$, where $\mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{f}''(\mathbf{x})$). Thus the quadratic convergence would be prevented.

If the non-linear problem has both equality and inequality constraints,

$$\begin{aligned} \mathbf{x}^* = \;& \text{argmin}_{\mathbf{x}\in\mathcal{P}} f(\mathbf{x}) \,, \\ \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n \mid\; & c_j(\mathbf{x}) = 0 \,, \;\; j = 1, \ldots, r \\ & c_j(\mathbf{x}) \geq 0 \,, \;\; j = r{+}1, \ldots, m\} \,, \end{aligned} \qquad (3.24)$$

then we can still use (3.23) except that the feasible region has to be changed accordingly. Thus the QO problem becomes

$$\begin{aligned}
&\text{Find } \mathbf{h} = \operatorname{argmin}_{\mathbf{h}\in\mathcal{P}_{\text{lin}}}\{q(\mathbf{h})\} \\
&\quad q(\mathbf{h}) = \tfrac{1}{2}\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{x},\boldsymbol{\lambda})\,\mathbf{h} + \mathbf{f}'(\mathbf{x})^\top \mathbf{h} + f(\mathbf{x})\,, \\
&\quad \mathcal{P}_{\text{lin}} = \{\mathbf{h}\in\mathbf{R}^n \mid c_j(\mathbf{x}) + \mathbf{c}'_j(\mathbf{x})^\top \mathbf{h} = 0\,,\ \ j=1,\ldots,r \\
&\qquad\qquad\qquad\qquad c_j(\mathbf{x}) + \mathbf{c}'_j(\mathbf{x})^\top \mathbf{h} \geq 0\,,\ \ j=r{+}1,\ldots,m\}
\end{aligned} \tag{3.25}$$

In applications the demand for second derivatives (in $\mathbf{L}''_{xx}(\mathbf{x},\boldsymbol{\lambda})$) can be an obstacle, and we may have to use approximations to these. Another problem with the method is that the quadratic model is good only for small values of $\|\mathbf{h}\|$. Therefore, when the current $\mathbf{x}$ is far away from the solution, it may be a good idea to retain the direction of $\mathbf{h}$ but reduce its length. In Section 4.2 we present a method where $\mathbf{L}''_{xx}(\mathbf{x},\boldsymbol{\lambda})$ is approximated by BFGS updating, and where a line search is incorporated in order to make the convergence robust also far from the solution.

Example 3.3. Consider the problem

$$f(\mathbf{x}) = x_1^2 + x_2^2, \quad \mathcal{P} = \{\mathbf{x}\in\mathbf{R}^2 \mid x_1^2 - x_2 - 1 = 0\} \tag{3.26}$$

The cost function is a quadratic in $\mathbf{x}$, but the constraint $c_1(\mathbf{x}) = x_1^2 - x_2 - 1$ is not linear, so this is **not** a quadratic optimization problem.

In Example 4.5 we solve this problem via a series of approximations of the form

$$\begin{aligned}
f(\mathbf{x}{+}\boldsymbol{\delta}) &\simeq q(\boldsymbol{\delta}) \equiv \tfrac{1}{2}\boldsymbol{\delta}^\top \mathbf{W}\,\boldsymbol{\delta} + \mathbf{f}'(\mathbf{x})^\top \boldsymbol{\delta} + \mathbf{f}(\mathbf{x}), \\
c(\mathbf{x}{+}\boldsymbol{\delta}) &\simeq l(\boldsymbol{\delta}) \equiv \mathbf{c}'_{\mathbf{1}}(\mathbf{x})^\top \boldsymbol{\delta} + c_1(\mathbf{x})\ \ ,
\end{aligned}$$

where q is the function of (3.23) with $\mathbf{L}''_{xx}(\mathbf{x},\boldsymbol{\lambda})$ replaced by an approximation $\mathbf{W}$. This leads to the following subproblem,

$$\begin{aligned}
\text{Find} \quad \mathbf{h} &= \operatorname{argmin}_{\boldsymbol{\delta}\in\mathcal{P}_{\text{lin}}}\{q(\boldsymbol{\delta})\} \\
\mathcal{P}_{\text{lin}} &= \{\mathbf{x}\in\mathbf{R}^2 \mid l(\boldsymbol{\delta}) = 0\}\ .
\end{aligned} \tag{3.27}$$

Let the first approximation for solving (3.26) correspond to $\mathbf{x} = [1,\ 1]^\top$ and $\mathbf{W} = \mathbf{I}$. Then

$$\begin{aligned}
q(\boldsymbol{\delta}) &= 2 + 2\delta_1 + 2\delta_2 + \tfrac{1}{2}\delta_1^2 + \tfrac{1}{2}\delta_1^2 \\
&= \tfrac{1}{2}(\delta_1 + 2)^2 + \tfrac{1}{2}(\delta_2 + 2)^2\ \ , \\
l(\boldsymbol{\delta}) &= -1 + 2\delta_1 - \delta_2\ \ .
\end{aligned}$$

The level curves of q are concentric circles centered at $\boldsymbol{\delta} = [-2,\ -2]^\top$, and the solution $\boldsymbol{\delta} = \mathbf{h}$ is the point, where one of these circles touches the line $l(\boldsymbol{\delta}) = 0$, see Figure 3.3a. The solution is $\mathbf{h} = [\,-0.8,\ -2.6\,]^\top$.

Using the line search to be described in Section 4.2 the next approximation is $\mathbf{x} := \mathbf{x} + \alpha\mathbf{h} = [0.620,\ -0.236]^\top$. This leads to the next quadratic approximation (3.27) with

$$\begin{aligned}
&\mathbf{f}'(\mathbf{x}) = \begin{bmatrix} 1.239 \\ -0.472 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 0.943 & -0.044 \\ -0.044 & 2.014 \end{bmatrix}\ , \\
&l(\boldsymbol{\delta}) = -0.380 + 1.238\delta_1 - \delta_2
\end{aligned}$$

where $\mathbf{W}$ is an updated approximation to $\mathbf{L}''_{xx}(\mathbf{x},\boldsymbol{\lambda})$ (see Section 4.2). The contours of q are concentric ellipses centered at $-\mathbf{W}^{-1}\mathbf{f}'(\mathbf{x}) = [-1.305,\ 0.206]^\top$ (the unconstrained minimizer of q, cf (3.5)).

Figure 3.3 shows the contours of q (full line) and f (dashed line) through the points given by $\boldsymbol{\delta} = \mathbf{0}$ and $\boldsymbol{\delta} = \mathbf{h}$. In the second case we see that in the region of interest q is a much better approximation to f than in the first case. Notice the difference in scaling and that each plot has the origin at the current $\mathbf{x}$. At the point $\mathbf{x} := \mathbf{x} + \alpha\mathbf{h} \simeq [\,0.702,\ -0.508\,]^\top$ we get $c_1(\mathbf{x}) \simeq 1.3{\cdot}10^{-4}$. This value is too small to be seen in Figure 3.3b.

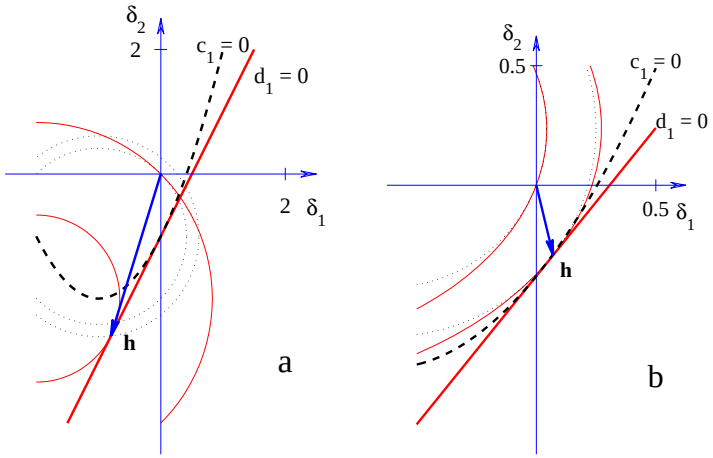


Figure 3.3: *Approximating quadratics and the constraint*
Dotted line: f and c. Full line: q and d

■

4. PENALTY AND SQO METHODS

There are several strategies on which to base methods for general constrained optimization. The first is called *sequential linear optimization*: in each iteration step we solve a linear optimization problem where both cost function and constraint functions are approximated linearly. This strategy may be useful e.g. in large scale problems.

The next strategy is *sequential quadratic optimization* (SQO). We introduced this in Section 3.4, and in section 4.2 we shall complete the description, including features that make it practical and robust.

The third strategy could be called *sequential unconstrained optimization* (SUO). In each iteration step we solve an unconstrained optimization problem, with the cost function modified to induce or force the next iterate to be feasible. The modification consists in adding a *penalty term* to the cost function. The penalty term is zero, if we are in the feasible region, and positive if we are outside it. The following examples are due to Fletcher (1993).

Example 4.1. Find

$$\operatorname{argmin}_{\mathbf{x}\in\mathcal{P}} f(\mathbf{x}), \qquad \mathcal{P} = \{\mathbf{x}\in\mathbf{R}^2 \mid c_1(\mathbf{x}) = 0\},$$

where

$$f(\mathbf{x}) = -x_1 - x_2, \qquad c_1(\mathbf{x}) = 1 - x_1^2 - x_2^2.$$

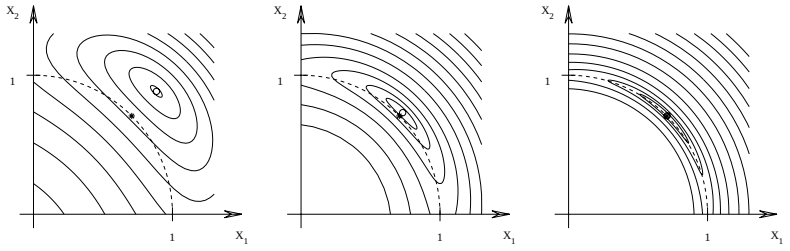
It is easy to see, that the solution is $\mathbf{x}^* = \frac{1}{\sqrt{2}}[1,\ 1]^\top$.

We penalize infeasible vectors by using the following function

$$\varphi(\mathbf{x},\sigma) = f(\mathbf{x}) + \tfrac{1}{2}\sigma\cdot(c_1(\mathbf{x}))^2,$$

where σ is a positive parameter. The penalty is zero if $\mathbf{x}$ is in $\mathcal{P}$, and positive

otherwise. In the case $\sigma = 0$ we have an unconstrained and unbounded problem: When the components of $\mathbf{x}$ tend to infinity, $f(\mathbf{x})$ tends to $-\infty$. In Figure 4.1 we see the contours of $\varphi(\mathbf{x}, 1)$, $\varphi(\mathbf{x}, 10)$ and $\varphi(\mathbf{x}, 100)$. For $\sigma > 0$ we have a minimizer $\mathbf{x}_\sigma$ and the figures indicate the desired convergence: $\mathbf{x}_\sigma \to \mathbf{x}^*$ for $\sigma \to \infty$.



$\sigma = 1$ $\sigma = 10$ $\sigma = 100$

Figure 4.1: *Contours and minimizer of* $\varphi(\mathbf{x}, \sigma)$ $\mathbf{x}^*$ *and* $\mathbf{x}_\sigma$ *is marked by* * *and* o*, respectively*

The figure indicates a very serious problem connected with SUO. As $\sigma \to \infty$, the valley around $\mathbf{x}_\sigma$ becomes longer and narrower making trouble for the method used to find this unconstrained minimizer. Another way of expressing this, is that the unconstrained problems become increasingly ill-conditioned. ■

Example 4.2. Consider the same problem as before, except that now c_1 is an inequality constraint: Find

$$\text{argmin}_{\mathbf{x}\in\mathcal{P}}\, f(\mathbf{x})\,, \qquad \mathcal{P} = \{\mathbf{x} \in \mathbf{R}^2 \mid c_1(\mathbf{x}) \geq 0\}\,,$$

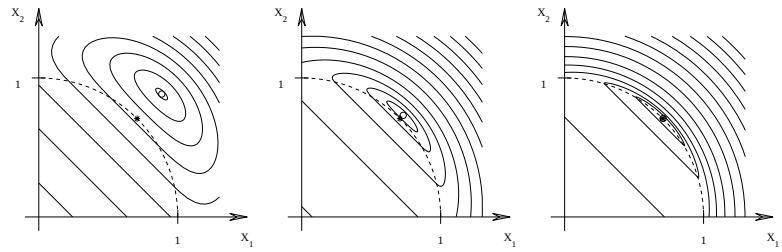
where f and c_1 are given in Example 4.1. The feasible region is the interior of the unit circle, and again the solution is $\mathbf{x}^* = \frac{1}{\sqrt{2}}[1,\ 1]^\top$.

The penalty term should reflect that all $\mathbf{x}$ for which $c_1(\mathbf{x}) \geq 0$ are permissible, and we can use

$$\varphi(\mathbf{x}, \sigma) = f(\mathbf{x}) + \tfrac{1}{2}\sigma\ (\min\{c_1(\mathbf{x}),\ 0\})^2\ , \quad \sigma \geq 0\,.$$

In Figure 4.2 we see the contours of $\varphi(\mathbf{x}, \sigma)$ and their minimizers $\mathbf{x}_\sigma$ for the same σ-values as in Example 4.1.

All the $\mathbf{x}_\sigma$ are infeasible and seem to converge to the solution. We still have the long narrow valleys and ill conditioned problems, when σ is large. With in-



$\sigma = 1$ $\sigma = 10$ $\sigma = 100$

Figure 4.2: *Contours and minimizer of* $\varphi(\mathbf{x}, \sigma)$ $\mathbf{x}^*$ *and* $\mathbf{x}_\sigma$ *is marked by* * *and* o*, respectively*

equality constraints there is an extra difficulty with this penalty function: Inside the feasible region the functions f and φ have the same values and derivatives, while this is not the case in the infeasible region. On the border of $\mathcal{P}$ (where the solution is situated) there is a discontinuity in the second derivative of $\varphi(\mathbf{x}, \sigma)$, and this disturbs line searches and descent directions which are based on interpolation, thus adding to the problems caused by the narrow valley. ■

It is characteristic for penalty methods, as indicated in the examples, that (normally) all the iterates are infeasible with respect to (some of) the inequality constraints. Therefore they are also called *exterior point methods*.

In some cases the objective function is undefined in (part of) the infeasible region. Then the use of exterior point methods becomes impossible. This has lead to the class of *barrier methods* that force all the iterates to be feasible. To contrast them with penalty function methods they are called *interior point methods* (IPM).

The most widely used IPMs are based on the *logarithmic barrier function*. We can illustrate it with a problem with one inequality constraint only,

$$\mathbf{x}^+ = \text{argmin}_{\mathbf{x}\in\mathcal{P}}\, f(\mathbf{x})\,, \qquad \mathcal{P} = \{\mathbf{x} \in \mathbf{R}^n \mid c_1(\mathbf{x}) \geq 0\}\,.$$

The corresponding barrier function is[1)]

$$\varphi(\mathbf{x}, \mu) = f(\mathbf{x}) - \mu\ \log c_1(\mathbf{x})\ ,$$

1) "log" is the natural (or Naperian) logarithm.

with the *barrier parameter* $\mu > 0$. The logarithm is defined only for $\mathbf{x}$ strictly inside $\mathcal{P}$ (we confine ourselves to working with real numbers), and since $\log c_1(\mathbf{x}) \to -\infty$ for $c_1(\mathbf{x}) \to 0$, we see that $\varphi(\mathbf{x}, \mu) \to +\infty$ for $\mathbf{x}$ approaching the border of $\mathcal{P}$. However, when $\mu \to 0$, the minimizer $\mathbf{x}_\mu$ of $\varphi(\mathbf{x}, \mu)$ can approach a point at the border.

Methods based on barrier functions share some of the disadvantages of the penalty function methods: As we approach the solution the intermediate results $\mathbf{x}_\mu$ are minimizers situated at the bottom of valleys that are narrow, ie $\mathbf{x}_\mu$ is the solution of an ill-conditioned (unconstrained) problem.

As indicated barrier methods are useful in problems where infeasible $\mathbf{x}$ vectors must not occur, but apart from this they may also be efficient in large scale problems. In linear optimization a number of very efficient versions have been developed during the 1990s, see eg Chapter 3 in Nielsen (1999).

We end this introduction by returning to the penalty functions used in Examples 4.1 and 44.2 and taking a look at the curvatures of the penalty function near the solution $\mathbf{x}^*$ and $\mathbf{x}_\sigma$, the unconstrained minimizer of $\varphi(\mathbf{x}, \sigma)$. Consider one inequality constraint as in Example 4.2, and assume that the constraint is *strongly active* at the solution: $\mathbf{f}'(\mathbf{x}^*) \neq \mathbf{0}$. This shows that

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}^*, \sigma) \neq \mathbf{0} \ ,$$

independent of σ, while the unconstrained minimizer $\mathbf{x}_\sigma$ satisfies

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}_\sigma, \sigma) = \mathbf{0} \ .$$

When $\sigma \to \infty$, $\mathbf{x}_\sigma \to \mathbf{x}^*$, but the difference in the gradients of φ (at $\mathbf{x}^*$ and $\mathbf{x}_\sigma$) remains constant, and thus the curvature of φ goes to infinity. This discrepancy is eliminated in the following method which was first introduced by Powell (1969).

4.1. The Augmented Lagrangian Method

At first we consider the special case where *only equality constraints* are present:

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{c}(\mathbf{x}) = \mathbf{0}\} \ ,$$

$\mathbf{c}$ being the vector function $\mathbf{c} : \mathbb{R}^n \mapsto \mathbb{R}^r$, whose ith component is the ith constraint function c_i. At the end of this section we generalize the formulation to include inequality constraints as well.

We have the following Lagrangian function (Definition 2.3),

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{c}(\mathbf{x}) \ ,$$

and introduce a penalty term as indicated at the beginning of this chapter. Thus consider the following *augmented Lagrangian function*[2)]

$$\varphi(\mathbf{x}, \boldsymbol{\lambda}, \sigma) = f(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{c}(\mathbf{x}) + \tfrac{1}{2}\sigma \, \mathbf{c}(\mathbf{x})^\top \mathbf{c}(\mathbf{x}) \ . \tag{4.1}$$

Notice that the discrepancy mentioned above has been relaxed: If $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, then the first order conditions in Corollary 2.6 and the fact that $\mathbf{c}(\mathbf{x}^*) = \mathbf{0}$ implies that $\mathbf{x}^*$ is a stationary point of φ:

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \sigma) = \mathbf{0} \ .$$

Furthermore, Fletcher has shown the existence of a finite number $\widehat{\sigma}$ with the property that if $\sigma > \widehat{\sigma}$, then $\mathbf{x}^*$ is an unconstrained local minimizer of $\varphi(\mathbf{x}, \boldsymbol{\lambda}^*, \sigma)$, ie if

$$\mathbf{x}_{\lambda,\sigma} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \varphi(\mathbf{x}, \boldsymbol{\lambda}, \sigma) \ , \tag{4.2}$$

then[3)]

$$\mathbf{x}_{\lambda^*,\sigma} = \mathbf{x}^* \quad \text{for all } \ \sigma > \widehat{\sigma} \ . \tag{4.3}$$

This means that the penalty parameter σ does not have to go to infinity. If σ is sufficiently large and if we insert $\boldsymbol{\lambda}^*$ (the vector of Lagrangian multipliers at the solution $\mathbf{x}^*$), then the unconstrained minimizer of the augmented Lagrangian function solves the constrained problem. Thus the problem of finding $\mathbf{x}^*$ has been reduced – or rather changed – to that of finding $\boldsymbol{\lambda}^*$.

We shall describe a method that uses the augmented Lagrangian function to find the solution. The idea is to use the penalty term to get close to the

---

[2)] Remember that $\boldsymbol{\lambda}^\top \mathbf{c}(\mathbf{x}) = \sum_{i=1}^m \lambda_i c_i(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})^\top \mathbf{c}(\mathbf{x}) = \sum_{i=1}^m (c_i(\mathbf{x}))^2$.

[3)] In case of several local minimizers "$\operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n}$" is interpreted as the local unconstrained minimizer in the valley around $\mathbf{x}^*$.

solution $\mathbf{x}^*$, and then let the Lagrangian term provide the final convergence by letting $\boldsymbol{\lambda}$ approach $\boldsymbol{\lambda}^*$. A rough sketch of the algorithm is

Choose initial values for $\boldsymbol{\lambda},\ \sigma$
repeat
  Compute $\mathbf{x}_{\lambda,\sigma}$
  Update $\boldsymbol{\lambda}$ and σ
until stopping criteria satisfied (4.4)

The computation of $\mathbf{x}_{\lambda,\sigma}$ (for fixed $\boldsymbol{\lambda}$ and σ) is an unconstrained optimization problem, which we deal with later. First, we concentrate on ideas for updating $(\boldsymbol{\lambda},\sigma)$ in such a way that σ stays limited and $\boldsymbol{\lambda}\to\boldsymbol{\lambda}^*$.

In the first iteration steps we keep $\boldsymbol{\lambda}$ constant (eg $\boldsymbol{\lambda}=\mathbf{0}$) and let σ increase. This should lead us close to $\mathbf{x}^*$ as described for penalty methods at the start of this chapter.

Next, we would like to keep σ fixed, $\sigma=\sigma_{\text{fix}}$, and vary λ. Then

$$\mathbf{x}_\lambda = \operatorname{argmin}_{\mathbf{x}\in\mathbf{R}^n}\ \varphi(\mathbf{x},\boldsymbol{\lambda},\sigma_{\text{fix}})$$

and

$$\psi(\boldsymbol{\lambda}) = \varphi(\mathbf{x}_\lambda,\boldsymbol{\lambda},\sigma_{\text{fix}}) = \min_{\mathbf{x}\in\mathbf{R}^n}\ \varphi(\mathbf{x},\boldsymbol{\lambda},\sigma_{\text{fix}})$$

are functions of $\boldsymbol{\lambda}$ alone. Assume $\sigma_{\text{fix}}>\widehat{\sigma}$. Since

1° $\psi(\boldsymbol{\lambda})$ is the minimal value of φ,

2° the definition (4.1) combined with $\mathbf{c}(\mathbf{x}^*)=\mathbf{0}$ shows that $\varphi(\mathbf{x}^*,\boldsymbol{\lambda},\sigma)=f(\mathbf{x}^*)$ for any $(\boldsymbol{\lambda},\sigma)$,

3° (4.3) implies $\mathbf{x}_{\lambda^*}=\mathbf{x}^*$,

it follows that for any $\boldsymbol{\lambda}$

$$\psi(\boldsymbol{\lambda}) \le \varphi(\mathbf{x}^*,\boldsymbol{\lambda},\sigma_{\text{fix}}) = \varphi(\mathbf{x}^*,\boldsymbol{\lambda}^*,\sigma_{\text{fix}}) = \psi(\boldsymbol{\lambda}^*)\ . \tag{4.5}$$

Thus the Lagrangian multipliers at the solution is a *local maximizer* for ψ,

$$\boldsymbol{\lambda}^* = \operatorname{argmax}_{\boldsymbol{\lambda}}\ \psi(\boldsymbol{\lambda})\ . \tag{4.6}$$

From the current $\boldsymbol{\lambda}$ we seek a step $\boldsymbol{\eta}$ such that $\boldsymbol{\lambda}+\boldsymbol{\eta}\simeq\boldsymbol{\lambda}^*$. In order to get a guideline on how to choose $\boldsymbol{\eta}$ we look at the Taylor expansion for ψ,

$$\begin{aligned}\psi(\boldsymbol{\lambda}+\boldsymbol{\eta}) &= \psi(\boldsymbol{\lambda}) + \boldsymbol{\eta}^\top\boldsymbol{\psi}'(\boldsymbol{\lambda}) + \tfrac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\psi}''(\boldsymbol{\lambda})\,\boldsymbol{\eta} + O(\|\boldsymbol{\eta}\|^3)\\ &= \psi(\boldsymbol{\lambda}) - \boldsymbol{\eta}^\top\mathbf{c} - \tfrac{1}{2}\boldsymbol{\eta}^\top\mathbf{J}_{\mathbf{c}}\,(\boldsymbol{\varphi}''_{\mathbf{xx}})^{-1}\,\mathbf{J}_{\mathbf{c}}^\top\,\boldsymbol{\eta} + O(\|\boldsymbol{\eta}\|^3)\ ,\end{aligned} \tag{4.7}$$

where $\mathbf{c}=\mathbf{c}(\mathbf{x}_\lambda)$, $\mathbf{J}_c=\mathbf{J}_c(\mathbf{x}_\lambda)$ is the Jacobian matrix defined in (3.21c), and $\boldsymbol{\varphi}''_{\mathbf{xx}}=\boldsymbol{\varphi}''_{\mathbf{xx}}(\mathbf{x}_\lambda,\boldsymbol{\lambda},\sigma_{\text{fix}})$. A proof of these expressions for the first and second derivatives of ψ can be found in Fletcher (1993). This expansion shows that

$$\boldsymbol{\eta} = -\alpha\,\mathbf{c}(\mathbf{x}_\lambda)\ ,\quad \alpha>0$$

is a step in the *steepest ascent direction*. Another way to get this, and at the same time providing a value for α, goes as follows: The vector $\mathbf{x}_\lambda$ is a minimizer for φ. Therefore $\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}_\lambda,\boldsymbol{\lambda},\sigma_{\text{fix}})=\mathbf{0}$, implying that

$$\mathbf{f}'(\mathbf{x}_\lambda) - \mathbf{J}_c(\mathbf{x}_\lambda)[\boldsymbol{\lambda}-\sigma_{\text{fix}}\mathbf{c}(\mathbf{x}_\lambda)] = \mathbf{0}\ .$$

Combining this with the KKT condition (Theorem 2.5),

$$\mathbf{f}'(\mathbf{x}^*) - \mathbf{J}_c(\mathbf{x}^*)\boldsymbol{\lambda}^* = \mathbf{0}\ ,$$

and the assumption that $\mathbf{x}_\lambda\simeq\mathbf{x}^*$, we find

$$\boldsymbol{\lambda}^* \simeq \boldsymbol{\lambda} - \sigma_{\text{fix}}\mathbf{c}(\mathbf{x}_\lambda)\ . \tag{4.8}$$

The right-hand side can be used for updating $\boldsymbol{\lambda}$. Fletcher (1993) shows that under certain regularity assumptions (4.8) provides linear convergence[4]. Faster convergence is obtained by applying Newton's method to the nonlinear problem $\boldsymbol{\psi}'(\boldsymbol{\lambda})=\mathbf{0}$,

$$\boldsymbol{\lambda}^* \simeq \boldsymbol{\lambda}+\boldsymbol{\eta}\ ,\quad \text{where}\quad \boldsymbol{\psi}''(\boldsymbol{\lambda})\boldsymbol{\eta} = -\boldsymbol{\psi}'(\boldsymbol{\lambda})\ .$$

Notice, that this is equivalent to finding $\boldsymbol{\eta}$ as a stationary point for the quadratic model obtained by dropping the error term $O(\|\boldsymbol{\eta}\|^3)$ in (4.7). A formula for $\boldsymbol{\psi}''(\boldsymbol{\lambda})$ is also given in (4.7). Inserting this we obtain

[4] This means that in the limit we have $\|\boldsymbol{\lambda}_{\text{new}}-\boldsymbol{\lambda}^*\|\le\kappa\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\|$, where $\boldsymbol{\lambda}_{\text{new}}=\boldsymbol{\lambda}-\sigma_{\text{fix}}\mathbf{c}(\mathbf{x}_\lambda)$ and $0<\kappa<1$.

$$\begin{aligned}\boldsymbol{\lambda}^* &\simeq \boldsymbol{\lambda} - [\boldsymbol{\psi}''(\boldsymbol{\lambda})]^{-1}\boldsymbol{\psi}'(\boldsymbol{\lambda}) \\ &= \boldsymbol{\lambda} - [\mathbf{J}_c\,(\boldsymbol{\varphi}''_{\mathbf{xx}})^{-1}\,\mathbf{J}_c^\top]^{-1}\mathbf{c}(\mathbf{x}_\lambda)\;. \end{aligned} \tag{4.9}$$

If the last expression of (4.9) is used for updating $\boldsymbol{\lambda}$ then quadratic convergence is obtained under certain regularity conditions, see Fletcher (1993). Notice that if a Quasi-Newton method is used in the unconstrained optimization for finding $\mathbf{x}_\lambda$ then an estimate of the inverse Hessian $(\boldsymbol{\varphi}''_{\mathbf{xx}})^{-1}$ is available.

Now we can present a specific example of an implementation of the algorithm outlined in (4.4). The details of course could be chosen in many other ways.

Algorithm 4.10. Augmented Lagrangian method
(Equality constraints only).

begin
  $k := 0$; $\mathbf{x} := \mathbf{x}_0$; $\boldsymbol{\lambda} := \boldsymbol{\lambda}_0$; $\sigma := \sigma_0$ $\{1^\circ\}$
  $K_{\text{prev}} := \|\mathbf{c}(\mathbf{x})\|_\infty$ $\{2^\circ\}$
  repeat
    $k := k+1$
    $\mathbf{x} := \text{argmin}_{\mathbf{x}}\varphi(\mathbf{x},\boldsymbol{\lambda},\sigma)$; $K := \|\mathbf{c}(\mathbf{x})\|_\infty$ $\{2^\circ\}$
    if $(K \le \frac{1}{4}K_{\text{prev}})$
      $\lambda := \text{Update}(\mathbf{x},\boldsymbol{\lambda},\sigma)$ $\{3^\circ\}$
      $K_{\text{prev}} := K$
    else
      $\sigma := 10 * \sigma$ $\{4^\circ\}$
  until $K < \varepsilon$ **or** $k > k_{\max}$
end

We have the following remarks:

1° As mentioned earlier it is natural to start with the pure penalty method, ie we let $\lambda_0 = \mathbf{0}$. σ_0 must be a positive number, one might eg start with $\sigma_0 = 1$. $\mathbf{x}_0$ is an initial estimate of the solution provided by the user.

2° K is meant to measure, how well the constraints are satisfied, and is used in the stopping criterion. A better measure (which can not be used as long as $\boldsymbol{\lambda} = \mathbf{0}$) is to take $K = \max_i |\lambda_i c_i(\mathbf{x})|$.

$\mathbf{x}$ is the minimizer of an unconstrained optimization problem, to be solved eg by one of the iterative methods given in Frandsen et al (2004). We assume that it can exploit "warm starts" (since after the first few iteration steps the new $\mathbf{x} = \mathbf{x}_{\lambda,\sigma}$ will be close to the previous one).

3° If K was reduced by 75% , then $\boldsymbol{\lambda}$ is updated by means of (4.8) or (4.9). Otherwise ...

4° ... we assume that $\mathbf{x}$ is too far from $\mathbf{x}^*$ and increase the penalty factor σ.

Example 4.3. We illustrate Algorithm 4.10 with the following simple problem, with $n = 2$ and $r = m = 1$:

minimize $f(\mathbf{x}) = x_1^2 + x_2^2$

with the constraint $c_1(\mathbf{x}) = 0\,, \quad c_1(\mathbf{x}) = x_1^2 - x_2 - 1\;.$

For hand calculation the following expressions are useful:

$$\mathbf{f}'(\mathbf{x}) = [\,2x_1,\ 2x_2\,]^\top, \qquad \mathbf{J}_c(\mathbf{x}) = \begin{bmatrix} 2x_1 & -1 \end{bmatrix}\;,$$

$$\varphi(\mathbf{x},\boldsymbol{\lambda},\sigma) = (x_1^2+x_2^2) - \boldsymbol{\lambda}\cdot(x_1^2 - x_2 - 1) + \sigma\cdot\left(x_1^2 - x_2 - 1\right)^2\;,$$

$$\varphi'_x(\mathbf{x},\boldsymbol{\lambda},\sigma) = \begin{bmatrix} 2x_1(1-\lambda+\sigma(x_1^2 - x_2 - 1)) \\ 2x_2 + \lambda - \sigma(x_1^2 - x_2 - 1) \end{bmatrix}\;,$$

$$\varphi''_{xx}(\mathbf{x},\boldsymbol{\lambda},\sigma) = \begin{bmatrix} 2x_1(1-\lambda-\sigma(x_2+1-3x_1^2)) & -2\sigma x_1 \\ -2\sigma x_1 & 2+\sigma \end{bmatrix}\;.$$

We shall follow the iterations from the starting point $\mathbf{x}_0 = [\,1,\ 1\,]^\top$, $\lambda_0 = 0$, $\sigma_0 = 2$. We find $K_{\text{prev}} = |c_1(\mathbf{x}_0)| = 1$.

First step: The augmented Lagrangian function is

$$\varphi(\mathbf{x},\boldsymbol{\lambda},\sigma) = (x_1^2+x_2^2) - 0\cdot(x_1^2 - x_2 - 1) + 1\cdot\left(x_1^2 - x_2 - 1\right)^2\;,$$

whose contours are shown below together with the minimizer, $\mathbf{x} = [\,0,\ -0.5\,]^\top$.

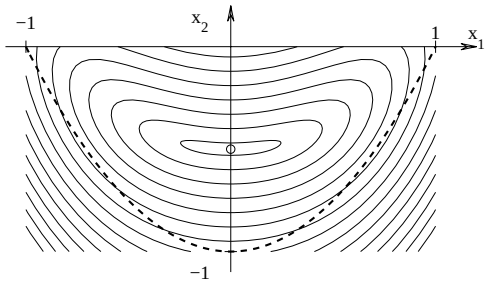


Figure 4.3: *Contours and minimizer of* $\varphi(\mathbf{x}, 0, 2)$. *The constraint* $c_1(\mathbf{x}) = 0$ *is dashed*

We get $K = |0^2 + 0.5 - 1| = 0.5$. Thus, K was reduced by less than 75 % and therefore we enter the `else` branch of Algorithm 4.10. The values for the next iteration step are $\lambda = 0,\ \sigma = 20$.

Second step: The augmented Lagrangian function is

$$\varphi(\mathbf{x}, 0, 20) = (x_1^2 + x_2^2) - 0 \cdot (x_1^2 - x_2 - 1) + 10 \cdot \left(x_1^2 - x_2 - 1\right)^2 .$$

There are two minimizers, and we assume that we find the minimizer with positive x_1: $\mathbf{x} = [\,\sqrt{0.45},\ -0.50\,]^\top$, $c_1(\mathbf{x}) = -0.05$, thus $K = 0.05$. This makes us enter the `if` branch: we will update the Lagrange factor. The steepest ascent method gives

$$\lambda := 0 - 20 \cdot (-0.05) = 1 ,$$

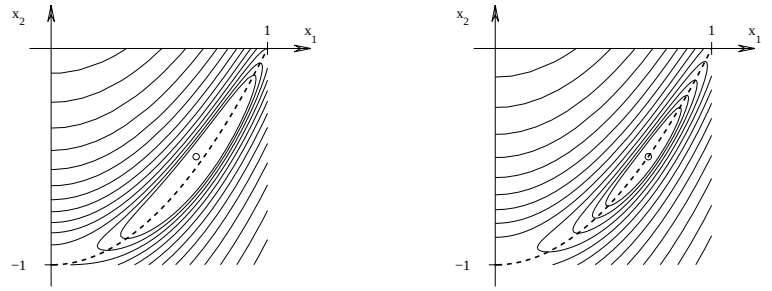
and this is also the result from the Newton method. The details are left as an exercise.

Third step:

$$\varphi(\mathbf{x}, 1, 20) = (x_1^2 + x_2^2) - 1 \cdot (x_1^2 - x_2 - 1) + 10 \cdot \left(x_1^2 - x_2 - 1\right)^2 ,$$

The minimizer is $\mathbf{x} = [\,\sqrt{0.5},\ -0.50\,]^\top \simeq [\,0.70711,\ -0.50\,]^\top$ with $K = 0$, so the algorithm stops. It has found the exact solution of the problem, $\mathbf{x}^* = \mathbf{x}$, and the corresponding Lagrangian multiplier is $\lambda^* = 1$, ie it is equal to the current λ-value. This exemplifies the comments on (4.3).

Below we give the contours of the augmented Lagrangian functions for steps two and three.



$\varphi(\mathbf{x}, 0, 20)$ $\varphi(\mathbf{x}, 1, 20)$

Figure 4.4: *Contours and minimizers of* $\varphi(\mathbf{x}, 0, 20)$ *and* $\varphi(\mathbf{x}, 1, 20)$, *respectively.* *The constraint* $c_1(\mathbf{x}) = 0$ *is dashed.* ∎

We now turn to the *general case,* where we have equality as well as inequality constraints,

$$\mathcal{P} = \{\mathbf{x} \in \mathbf{R}^n \mid \begin{array}{l} c_i(\mathbf{x}) = 0,\ i = 1, \ldots, r \\ c_i(\mathbf{x}) \geq 0,\ i = r{+}1, \ldots, m\} \end{array} .$$

4.1.1. An easy solution. A straight forward way to solve this problem would be to use the method just described: Let a penalty method bring us to the neighbourhood of a solution, and then simply consider the active or near active constraints as equality constraints. Discard the rest of the constraints (still keeping an eye on them, though, to observe whether they remain inactive), and use one of the two methods for updating the vector of Lagrange multipliers $\boldsymbol{\lambda}$.

The augmented Lagrangian function could be the following:

$$\widetilde{\varphi}(\mathbf{x}, \boldsymbol{\lambda}, \sigma) = f(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{d}(\mathbf{x}) + \tfrac{1}{2}\sigma \mathbf{d}(\mathbf{x})^\top \mathbf{d}(\mathbf{x}) ,$$

where $\mathbf{d}(\mathbf{x})$ is defined as follows

$$d_i(\mathbf{x}) = \begin{cases} c_i(\mathbf{x}) & \text{if } i \in \mathcal{A}_\delta(\mathbf{x}) , \\ 0 & \text{otherwise.} \end{cases}$$

Here, we have defined the approximate active set $\mathcal{A}_\delta(\mathbf{x})$ by

$$\mathcal{A}_\delta(\mathbf{x}) \;=\; \{1,\ldots,r\} \cup \{\; i \mid \; i > r \;\text{ and }\; c_i(\mathbf{x}) \le \delta \;\}\,, \tag{4.11}$$

where δ is a small positive number. Initially we could keep $\boldsymbol{\lambda} = 0$ and increase σ until the approximate active set seems to have stabilized (eg by being constant for two consecutive iterations). As long as $\mathcal{A}(\mathbf{x})$ remains constant we update $\boldsymbol{\lambda}$ using (4.8) or (4.9) (discarding inactive constraints and assuming that the active inequality constraints are numbered first). Otherwise $\boldsymbol{\lambda}$ is set to $\mathbf{0}$ and σ is increased.

The algorithm might be outlined as follows:

Algorithm 4.12. Augmented Lagrangian method
(General problem, easy solution).

begin
  $\boldsymbol{\lambda} := \mathbf{0}$; $\sigma := \sigma_0$
  repeat
    $\mathbf{x} := \text{argmin}_{\mathbf{x}} \widetilde{\varphi}(\mathbf{x}, \boldsymbol{\lambda}, \sigma)$
    if (stable active set $\mathcal{A}_\delta(\mathbf{x})$)
      $\lambda := \text{Update}(\mathbf{x}, \boldsymbol{\lambda}, \sigma)$
    else
      $\boldsymbol{\lambda} := 0$; $\sigma := 10 * \sigma$
  until STOP
end

Many alternatives for defining the active set could be considered. It might, eg, depend on the values of $|c_i(\mathbf{x})|,\ i = 1,\ ,...,\ m$. One disadvantage about this type of definition is that a threshold value, like δ, must be provided by the user. This might be avoided by a technique like the one in Algorithm 4.10 (and the following Algorithm 4.20).

4.1.2. A better solution. We change the inequality constraints ($i = r{+}1,\ldots,m$) into equality constraints by introducing so-called *slack variables* z_i:

$$c_{r+i}(\mathbf{x}) \ge 0 \quad \Leftrightarrow \quad \begin{cases} c_{r+i}(\mathbf{x}) - z_i = 0 \\ z_i \ge 0 \end{cases}\,,\ \ i = 1,\ldots,m{-}r\,. \tag{4.13}$$

Notice, that we have extended the number of variables, and still have inequality constraints. These are simple, however, and – as we shall see – the slack variables can be eliminated.

Consider the augmented Lagrangian function corresponding to the m equality constraints,

$$\begin{aligned}\varphi(\mathbf{x},\mathbf{z},\boldsymbol{\lambda},\sigma) = f(\mathbf{x}) &- \textstyle\sum_{i=1}^{r} \lambda_i c_i(\mathbf{x}) + \frac{1}{2}\sigma \sum_{i=1}^{r} c_i(\mathbf{x})^2 \\ &- \textstyle\sum_{i=r+1}^{m} \lambda_i (c_i(\mathbf{x}) - z_{i-r}) \\ &+ \textstyle\frac{1}{2}\sigma \sum_{i=r+1}^{m} (c_i(\mathbf{x}) - z_{i-r})^2\,. \end{aligned} \tag{4.14}$$

For fixed $\boldsymbol{\lambda}$ and σ we wish to find $\mathbf{x}_{\lambda,\sigma}$ and $\mathbf{z}_{\lambda,\sigma}$ that minimize φ under the constraint $\mathbf{z}_{\lambda,\sigma} \ge \mathbf{0}$. $\mathbf{x}_{\lambda,\sigma}$ minimizes the original problem provided that σ is sufficiently large and λ is the vector of Lagrange multipliers at the solution.

At the minimizer ($\mathbf{x}_{\lambda,\sigma}$,$\mathbf{z}_{\lambda,\sigma}$) either $z_i = 0$ (the constraint $z_i \ge 0$ is active) or $\dfrac{\partial\varphi}{\partial z_i} = 0$. Now, from (4.14) we see that

$$\frac{\partial\varphi}{\partial z_{i-r}} = \lambda_i - \sigma(c_i(\mathbf{x}) - z_{i-r})\,,$$

and equating this with zero we get $z_{i-r} = c_i(\mathbf{x}) - \dfrac{1}{\sigma}\,\lambda_i$. Thus, the relevant values for the slack variables are

$$z_{i-r} = \max\{0,\ c_i(\mathbf{x}) - \frac{1}{\sigma}\,\lambda_i\},\ \ i = r{+}1,\ldots,m\,.$$

Inserting this in (4.14) will make $\mathbf{z}$ disappear, and we obtain

$$\varphi(\mathbf{x},\boldsymbol{\lambda},\sigma) = f(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{d}(\mathbf{x}) + \tfrac{1}{2}\sigma \mathbf{d}(\mathbf{x})^\top \mathbf{d}(\mathbf{x})\,, \tag{4.15a}$$

where $\mathbf{d}(\mathbf{x})$ hold the modified equality constraint functions given by

$$d_i(\mathbf{x}) = \begin{cases} c_i(\mathbf{x}) & \text{if } \ i \le r \quad \text{or } \ c_i(\mathbf{x}) \le \frac{1}{\sigma}\,\lambda_i \\ \frac{1}{\sigma}\,\lambda_i & \text{otherwise} \end{cases}\,. \tag{4.15b}$$

Thus, the augmented Lagrangian function for the generally constrained problem is very similar to (4.1).

Letting

$$\psi(\boldsymbol{\lambda}) = \min_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}, \boldsymbol{\lambda}, \sigma_{\text{fix}}) = \varphi(\mathbf{x}_\lambda, \boldsymbol{\lambda}, \sigma_{\text{fix}}) \,, \tag{4.16}$$

the inequality $\psi(\boldsymbol{\lambda}) \le \psi(\boldsymbol{\lambda}^*)$ corresponding to (4.5), can easily be shown valid. Thus $\boldsymbol{\lambda}^*$ maximizes $\boldsymbol{\psi}$ so $\boldsymbol{\psi}'(\boldsymbol{\lambda}^*) = \mathbf{0}$.

The *steepest ascent* iteration, corresponding to (4.8), is

$$\boldsymbol{\lambda}_{sa} = \boldsymbol{\lambda} - \sigma_{\text{fix}}\mathbf{d}(\mathbf{x}_\lambda) \,. \tag{4.17}$$

The *Newton iteration* for solving $\boldsymbol{\psi}'(\boldsymbol{\lambda}^*) = \mathbf{0}$, corresponding to (4.9), is

$$\boldsymbol{\lambda}_{new} = \boldsymbol{\lambda} + \boldsymbol{\eta} \,, \quad \text{where} \quad \boldsymbol{\psi}''(\boldsymbol{\lambda})\boldsymbol{\eta} = -\boldsymbol{\psi}'(\boldsymbol{\lambda}) \,. \tag{4.18}$$

Here the first and second order derivatives of ψ are (see Fletcher (1993))

$$\begin{aligned} &\boldsymbol{\psi}'(\boldsymbol{\lambda}) = -\mathbf{d}(\mathbf{x}_\lambda) \,, \\ &\boldsymbol{\psi}''(\boldsymbol{\lambda}) = -\begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sigma}\mathbf{I} \end{bmatrix} \quad \text{with} \quad \mathbf{G} = \widetilde{\mathbf{J}}_c \left(\boldsymbol{\varphi}''_{xx}\right)^{-1} \widetilde{\mathbf{J}}_c^{\top} \,. \end{aligned} \tag{4.19}$$

In $\widetilde{\mathbf{J}}$ we only consider the active constraints (first line in (4.15b)) and we assume that these are numbered first. Thus $\mathbf{G}$ is an s by s matrix (where s is the number of active constraints), and $\mathbf{I}$ is the unit matrix in $\mathbb{R}^{m-s}$. Notice that if constraint number i is inactive at ($\mathbf{x}$,$\boldsymbol{\lambda}$) (last line in (4.15b)) then the value of η_i in (4.18) is $-\lambda_i$. Thus the i'th component of $\boldsymbol{\lambda}_{new}$ will be 0 which is consistent with remark 3° on Theorem 2.5.

The algorithm is given below. Essentially, it is identical with 4.10. We have the following remarks:

1° As remark 1° to Algorithm 4.10.

2° As remark 2° to Algorithm 4.10, except for K: For active constraints $|d_i(\mathbf{x})|$ is the deviation from $c_i(\mathbf{x})$ to zero. For an inactive constraint, $|d_i(\mathbf{x})| = |\lambda_i/\sigma|$ which becomes 0 when $\boldsymbol{\lambda}$ is updated. If this constraint is also inactive at the solution, then $\lambda_i^* = 0$, see remark 3° on Theorem 2.5; thus, also in this case the value $|d_i|$ is relevant for the stopping criterion.

Algorithm 4.20. Augmented Lagrangian method.
(General case).

begin
  $k := 0$; $\mathbf{x} := \mathbf{x}_0$; $\boldsymbol{\lambda} := \boldsymbol{\lambda}_0$; $\sigma := \sigma_0$ {1°}
  $K_{\text{prev}} := \|\mathbf{d}(\mathbf{x})\|_\infty$ {2°}
  repeat
    $k := k+1$
    $\mathbf{x} := \text{argmin}_{\mathbf{x}}\varphi(\mathbf{x}, \boldsymbol{\lambda}, \sigma)$; $K := \|\mathbf{d}(\mathbf{x})\|_\infty$ {2°}
    if $(K \le \frac{1}{4}K_{\text{prev}})$
      $\lambda := \text{Update}(\mathbf{x}, \boldsymbol{\lambda}, \sigma)$ {3°}
      $K_{\text{prev}} := \max(K, K_{\text{prev}})$
    else
      $\sigma := 10 * \sigma$
  until $K < \varepsilon$ **or** $k > k_{\max}$
end

3° The updating of $\boldsymbol{\lambda}$ can be made by the steepest ascent formula (4.17), which is efficient initially, or by Newton's method (4.18), which provides quadratic final convergence (under the usual regularity conditions).

If a Quasi-Newton method is used to find $\mathbf{x}$ at 3°, then an approximate $\boldsymbol{\varphi}''$ (or $(\boldsymbol{\varphi}'')^{-1}$) is available and can be used in (4.19b). In this case we do not obtain quadratic but superlinear convergence, which is almost as good.

Algorithm 4.20 has proved to be robust and quite efficient in practice. Typically the solution is found after 3 – 10 runs through the repeat loop. In Example 4.6 we report results of some test runs with the algorithm.

4.2. The Lagrange-Newton Method

In Section 3.4 we formulated the problem of finding a local constrained minimizer of f as a problem of finding a stationary point of the associated Lagrangian function. Applying Newton's method to this, we saw that each step was equivalent to a quadratic optimization problem (QO). The next Newton step gives rise to a new QO, and therefore the names *"Lagrange-Newton method"* and *"sequential quadratic optimization"* are more or less synonymous. The shorter name *"SQP"* is also used because a quadratic optimization problem is called a *"quadratic program"* in older literature.

The ideas date back to the 1960s, but the first efficient implementations were developed by Han (1976) and Powell (1977). Currently it is considered as the most efficient method (except for problems with extremely simple function evaluations (as all the problems in the examples of this booklet)).

We now complete the description of the method from Section 3.4 and introduce features that improve the global performance of the method. This includes soft line search with a special type of penalty function. We conclude the description with an update method for the Hessian matrix. This actually makes the method a Quasi-Newton method see Chapter 5 in Frandsen et al (2004)) with good final (superlinear) convergence without having to implement second derivatives, which would be needed with a true Newton's method (giving quadratic final convergence).

Summarizing (and slightly modifying) the description from Section 3.4, we can state the SQP method in algorithmic form

$$\begin{array}{l}\text{Choose } \mathbf{x}_0; \quad \mathbf{x} := \mathbf{x}_0\\ \textbf{repeat}\\ \quad \mathbf{h} := \operatorname{argmin}_{\boldsymbol{\delta}\in\widetilde{\mathcal{P}}}\, q(\boldsymbol{\delta})\\ \quad \text{Find step parameter } \alpha\\ \quad \mathbf{x} := \mathbf{x} + \alpha\mathbf{h}\\ \textbf{until}\ \text{stopping criteria satisfied}\end{array} \tag{4.21}$$

Here q is a quadratic model of the cost function in the neighbourhood of $\mathbf{x}$,

$$f(\mathbf{x}+\boldsymbol{\delta}) \simeq q(\boldsymbol{\delta}) = f(\mathbf{x}) + \boldsymbol{\delta}^\top \mathbf{f}\,'(\mathbf{x}) + \tfrac{1}{2}\boldsymbol{\delta}^\top \mathbf{W}(\mathbf{x})\boldsymbol{\delta}, \tag{4.22a}$$

and $\widetilde{\mathcal{P}}$ is the feasible region

$$\begin{aligned}\widetilde{\mathcal{P}} = \{\boldsymbol{\delta}\in\mathbf{R}^n \mid\ & d_i(\boldsymbol{\delta}) = 0,\ i=1,\dots,r\\ & d_i(\boldsymbol{\delta}) \ge 0,\ i=r+1,\dots,m\}\,,\end{aligned} \tag{4.22b}$$

corresponding to a linear model of the constraints

$$\mathbf{c}(\mathbf{x}+\boldsymbol{\delta}) \simeq \mathbf{d}(\boldsymbol{\delta}) = \mathbf{c}(\mathbf{x}) + \mathbf{J}_c(\mathbf{x})\boldsymbol{\delta}\,, \tag{4.22c}$$

where $(\mathbf{J}_c)_{ij} = \dfrac{\partial c_i}{\partial x_j}$, cf (3.21d).

We shall discuss the choice of the step parameter α in (4.21) and matrix $\mathbf{W}(\mathbf{x})$ in (4.22a). First, however, let us consider the consequences of the linearization (4.22c) of the constraint functions.

Example 4.4. We consider a problem in $\mathbf{R}^2$ with one inequality constraint only, $c_1(\mathbf{x}) \ge 0$. Figure 4.5 shows the border curve for the feasible region.

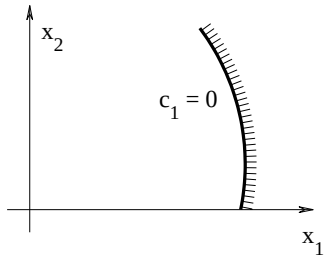


Figure 4.5: *Border curve of feasible region, $c_1(\mathbf{x}) = 0$. The infeasible side is hatched*

We want to study the variation of the function $c_1(\mathbf{x})$ around this border curve. Figure 4.6a shows the surface $y = c_1(\mathbf{x})$, and in Figure 4.6b we have added the tangent plane to this constraint surface at a point $(\mathbf{x}, c_1(\mathbf{x}))$, where $c_1(\mathbf{x}) > 0$.

The tangent plane corresponds to the linear approximation

$$c_1(\mathbf{x}+\mathbf{h}) \simeq \kappa(\mathbf{x}+\mathbf{h}) = c_1(\mathbf{x}) + \mathbf{h}^\top \mathbf{c}_1'(\mathbf{x})\,.$$

We assume that c_1 is strictly concave (so that the feasible region is convex, cf Theorem 1.18). Then the tangent plane at any point is above the constraint surface (except at the point of osculation), and a consequence is that the line $\psi(\mathbf{x}+\mathbf{h}) = 0$ is completely outside the feasible region. This is illustrated in

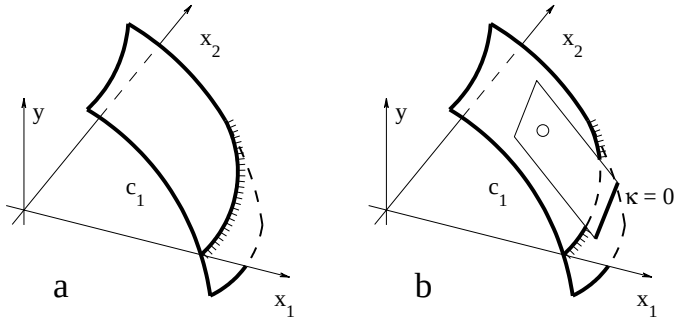


Figure 4.6: *Variation of the constraint function* $c_1(\mathbf{x})$ *near the border curve* $c_1(\mathbf{x})=0$ *and the tangent plane at the point* $(\mathbf{x}, c_1(\mathbf{x}))$*, marked by a circle.*

Figure 4.6b for $\mathbf{x}\in\mathcal{P}$, and it is easily seen that also if $\mathbf{x}$ is infeasible ($c_1(\mathbf{x})<0$), then the line $\psi(\mathbf{x}+\mathbf{h})=0$ is completely outside $\mathcal{P}$.

The properties described above are valid in general, except for cases, where the concave function has a local maximizer between $\mathbf{x}$ and the border curve. ∎

4.2.1. Choice of step length α. The solution $\mathbf{h}$ to the quadratic optimization problem in (4.21) satisfies all the linearized constraints, and, as shown in the previous example, this may cause $\mathbf{x}+\mathbf{h}$ to be infeasible with respect to the true constraints. Also, if $\mathbf{h}$ comes out too large, then the quadratic model may be a poor approximation to the true variation of the cost function. Therefore we make a line search similar to the soft line search described in Section 2.5 of Frandsen et al (2004). The function considered in the line search is a so-called "*exact penalty function*"[5)]

$$\pi(\mathbf{y},\boldsymbol{\mu}) = f(\mathbf{y}) + \sum_{i=1}^{r}\mu_i|c_i(\mathbf{x})| + \sum_{i=r+1}^{m}\mu_i|\min\{0,c_i(\mathbf{y})\}| \quad \text{with } \mu_i \geq |\lambda_i| \,. \tag{4.23}$$

The penalty factors are chosen as $\boldsymbol{\mu}=|\boldsymbol{\lambda}|$ in the first iteration step, while

[5)] This penalty function is exact in the sense that the solution $\mathbf{x}^*$ of our problem minimizes $\pi(\mathbf{y},\boldsymbol{\mu})$ for any $\boldsymbol{\mu}$ with $\boldsymbol{\mu}\geq\mathbf{0}$.

some conservatism is recommended in later steps,

$$\mu_i := \max\{|\lambda_i|,\ \tfrac{1}{2}(\mu_i+|\lambda_i|)\}\,. \tag{4.24}$$

This is specially important for constraints that are active in one iteration step and inactive in the next.

Powell has shown that the function

$$\pi(\alpha) = \pi(\mathbf{x}+\alpha\mathbf{h},\boldsymbol{\mu}) \quad \text{for } \alpha\geq 0,\ \mathbf{h} \text{ and } \boldsymbol{\mu} \text{ fixed}\,,$$

has $\pi'(0)<0$, so that a line search can lead to a point $\mathbf{x}+\alpha\mathbf{h}$, which is "better" in terms of this measure. Also, even for moderate penalty, the minimizer is exactly feasible.

The disadvantage of an exact penalty function is that it is not differentiable; $\pi(\alpha)$ has kinks at the points where a constraint $c_i(\mathbf{x}+\alpha\mathbf{h})$ passes the value of zero; see Figure 4.7 below.

We also need a piecewise linear approximation to $\pi(\alpha)$,

$$\begin{aligned}\pi(\alpha)\simeq\psi(\alpha) = f(\mathbf{x}) &+ \alpha\mathbf{h}^\top\mathbf{f}'(\mathbf{x})\\ &+\textstyle\sum_{i=1}^{r}\mu_i|c_i(\mathbf{x})+\alpha\mathbf{h}^\top\mathbf{c_i'}(\mathbf{x})|\\ &+\textstyle\sum_{i=r+1}^{m}\mu_i|\min\{0,\ c_i(\mathbf{x})+\alpha\mathbf{h}^\top\mathbf{c_i'}(\mathbf{x})\}|\end{aligned}$$

An example of $\pi(\alpha)$ and $\psi(\alpha)$ is shown in Figure 4.7. Notice that $\psi(\alpha)$ is convex, and it also has kinks, situated differently from the kinks of π.

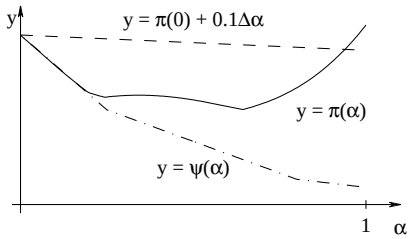


Figure 4.7: *Line search function* $\pi(\alpha)$ *and its linear approximation* $\psi(\alpha)$

Similar to a "normal" soft line search, we accept a value of α such that

the point $(\alpha, \pi(\alpha))$ is below the dashed line indicated in Figure 4.7. The slope of this line is 10% of the slope of the chord between $(0, \psi(0))$ and $(1, \psi(1))$, ie

$$\begin{aligned}\Delta &= \psi(1) - \psi(0) \\ &= \mathbf{h}^\top \mathbf{f}\,'(\mathbf{x}) - \sum_{i=1}^{r} \mu_i |c_i(\mathbf{x})| - \sum_{i=r+1}^{m} \mu_i |\min\{0,\ c_i(\mathbf{x})\}| \,. \end{aligned} \tag{4.25}$$

In this expression we have used the fact that $\psi(1) = \mathbf{h}^\top \mathbf{f}\,'(\mathbf{x})$ since the other terms are zero for $\mathbf{h} \in \widetilde{\mathcal{P}}$. Note that $\mathbf{h}$ is downhill for f, and therefore Δ is guaranteed to be negative.

In each step of the line search algorithm we use a second order polynomial $P(t)$ to approximate $\pi(t)$ on the interval $[0, \alpha]$. The coefficients are determined so that $P(0) = \pi(0)$, $P\,'(0) = \Delta$, $P(\alpha) = \pi(\alpha)$,

$$P(t) = \pi(0) + \Delta t + (\pi(\alpha) - \pi(0) - \Delta\alpha)\frac{t^2}{\alpha^2} \,.$$

If the coefficient to t^2 is positive, then this polynomial has a minimizer β, determined by $P'(\beta) = 0$, or

$$\beta = \frac{-\Delta\alpha^2}{2(\pi(\alpha) - \pi(0) - \Delta\alpha)} \,. \tag{4.26}$$

Now we can formulate the line search algorithm:

Algorithm 4.27. Penalty Line Search.

begin
  $\alpha := 1$; Compute Δ by (4.25)
  while $\pi(\alpha) \geq \pi(0) + 0.1\Delta\alpha$
    Compute β by (4.26)
    $\alpha := \min\{0.9\alpha,\ \max\{\beta,\ 0.1\alpha\}\ \}$
end

The expression for the new α ensures that the algorithm does not get stuck at the current value and, on the other hand, does not go to zero too fast. The algorithm has been validated by experience.

4.2.2. Choice of W in (4.22). By comparison with the Taylor expansion (1.6) an obvious choice is $\mathbf{W}(\mathbf{x}) = \mathbf{f}\,''(\mathbf{x})$. However, the goal is to find a minimizer for the Lagrangian function $L(\mathbf{x}, \boldsymbol{\lambda})$, and the description in Section 3.4 shows that a more appropriate choice is

$$\mathbf{W}(\mathbf{x}) \ = \ \mathbf{L}''_{xx}(\mathbf{x}, \boldsymbol{\lambda}) \ = \ \mathbf{f}\,''(\mathbf{x}) - \sum \lambda_i \mathbf{c}_i''(\mathbf{x}) \,.$$

We know from Theorem 2.11 that at the solution $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ the Hessian matrix satisfies $\mathbf{h}^\top \mathbf{L}''_{xx}(\mathbf{x}^*, \boldsymbol{\lambda}^*)\mathbf{h} \geq 0$ for all feasible directions. This does not imply that $\mathbf{L}''_{xx}(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is positive definite, but contributes to the theoretical motivation for the following strategy that has proven successful: Start with a positive definite $\mathbf{W}(\mathbf{x}_0)$, eg $\mathbf{W}(\mathbf{x}_0) = \mathbf{I}$. In each iteration step update $\mathbf{W}$ so that it is positive definite, thus giving a well-defined descent direction. The use of an updating strategy has the further benefit that we do not have to supply second derivatives of the cost function f and the constraint functions $\{c_i\}$.

A good updating strategy is the BFGS method discussed in Section 5.10 of Frandsen et al (2004). Given the current $\mathbf{W} = \mathbf{W}(\mathbf{x})$ and the next iterate $\mathbf{x}_{\text{new}} = \mathbf{x} + \alpha\mathbf{h}$. The change in the gradient of Lagrange's function (with respect to $\mathbf{x}$) is

$$\begin{aligned}\mathbf{y} &= \mathbf{L}'_x(\mathbf{x}_{\text{new}}, \boldsymbol{\lambda}) - \mathbf{L}'_x(\mathbf{x}, \boldsymbol{\lambda}) \\ &= \mathbf{f}\,'(\mathbf{x}_{\text{new}}) - \mathbf{f}\,'(\mathbf{x}) - (\mathbf{J}_c(\mathbf{x}_{\text{new}}) - \mathbf{J}_c(\mathbf{x}))^\top \boldsymbol{\lambda} \,. \end{aligned} \tag{4.28a}$$

We check the so-called *curvature condition*,

$$\mathbf{y}^\top(\mathbf{x}_{\text{new}} - \mathbf{x}) > 0 \,. \tag{4.28b}$$

If this is satisfied, then $\mathbf{W}$ is "positive definite with respect to the step direction $\mathbf{h}$", and so is $\mathbf{W}_{\text{new}}$ found by the BFGS formula,

$$\begin{aligned}&\mathbf{W}_{\text{new}} = \mathbf{W} + \frac{1}{\alpha\mathbf{h}^\top\mathbf{y}}\,\mathbf{y}\mathbf{y}^\top - \frac{1}{\mathbf{h}^\top\mathbf{u}}\,\mathbf{u}\mathbf{u}^\top \,, \\ &\text{where} \quad \mathbf{u} = \mathbf{W}\mathbf{h} \,. \end{aligned} \tag{4.28c}$$

If the curvature condition is not satisfied, then we let $\mathbf{W}_{\text{new}} = \mathbf{W}$.

4.2.3. Stopping criterion. We use the following measure for the goodness of the approximate solution obtained as $\mathbf{x} = \mathbf{x}_{\text{prev}} + \alpha\mathbf{h}$,

$$\eta(\mathbf{x}, \boldsymbol{\lambda}) = |q(\alpha\mathbf{h}) - f(\mathbf{x})| + \sum_{i\in\mathcal{B}} \lambda_i |c_i(\mathbf{x})| + \sum_{i\in\mathcal{J}} |\min\{0, c_i(\mathbf{x})\}| \,. \tag{4.29}$$

As in Chapter 3, $\mathcal{B}$ is the set of equality and active inequality constraints, and $\mathcal{J}$ is the set of inactive inequality constraints. The first term measures the quality of the approximating quadratic (4.22a) and the other terms measure how well the constraints are satisfied.

4.2.4. Summary. The algorithm can be summarized as follows. The parameters ε and $k_{\max}$ must be set by the user.

Algorithm 4.30. Lagrange – Newton Method.

begin
  $\mathbf{x} := \mathbf{x}_0$; $\mathbf{W} := \mathbf{W}_0$; $\boldsymbol{\mu} := \mathbf{0}$; $k := 0$
  repeat
    $k := k+1$
    Find $(\mathbf{h}, \boldsymbol{\lambda})$ by Algorithm 3.10
    Update $\boldsymbol{\mu}$ by (4.24)
    Find α by Algorithm 4.27
    $\mathbf{x}_{\text{prev}} := \mathbf{x}$; $\mathbf{x} := \mathbf{x} + \alpha\mathbf{h}$
    Update $\mathbf{W}$ by (4.28)
  until $\eta(\mathbf{x}, \boldsymbol{\lambda}) < \varepsilon$ **or** $k > k_{\max}$
end

Example 4.5. We shall use the algorithm on the same problem as in Example 4.3,

$$\text{minimize } f(\mathbf{x}) = x_1^2 + x_2^2$$
$$\text{with the constraint } c_1(\mathbf{x}) = 0\,, \quad c_1(\mathbf{x}) = x_1^2 - x_2 - 1\,,$$

and with the same starting point, $\mathbf{x}_0 = [\,1,\ 1\,]^\top$. Further, we choose $\mathbf{W}_0 = \mathbf{I}$.

We shall need the following expressions

$$\mathbf{f}'(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}, \qquad \mathbf{J}_c(\mathbf{x}) = \begin{bmatrix} 2x_1 & -1 \end{bmatrix},$$
$$q(\boldsymbol{\delta}) = f(\mathbf{x}) + 2(x_1\delta_1 + x_2\delta_2) + \tfrac{1}{2}\left(w_{11}\delta_1^2 + 2w_{12}\delta_1\delta_2 + w_{22}\delta_2^2\right)$$
$$d_1(\boldsymbol{\delta}) = c_1(\mathbf{x}) + 2x_1\delta_1 - \delta_2 \,.$$

The first model problem is

$$\text{minimize} \quad q(\boldsymbol{\delta}) = 2 + 2\delta_1 + 2\delta_2 + 0.5\delta_1^2 + 0.5\delta_1^2$$
$$\text{subject to} \quad d_1(\boldsymbol{\delta}) = -1 + 2\delta_1 - \delta_2 = 0\,.$$

This was discussed in Example 3.3, where we found the minimizer $\mathbf{h} = [-0.8,\ -2.6\,]^\top$. The corresponding Lagrange multiplier is $\lambda = 0.6$, and this is also used as the first value for the penalty parameter μ. Figure 4.8 shows

$$\pi(\alpha) = (1-.8\alpha)^2 + (1-2.6\alpha)^2 + .6|(1-.8\alpha)^2 - (1-2.6\alpha) - 1|\,.$$

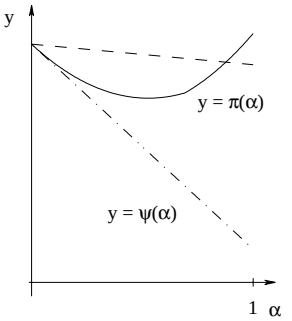


Figure 4.8: *Penalty function* $\pi(\alpha)$ *and linear approximation* $\psi(\alpha)$.

The linear approximation is

$$\psi(\alpha) = 2 + \alpha\begin{bmatrix} -.8 & -2.6 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \end{bmatrix} + .6|-1 + \alpha\begin{bmatrix} -.8 & -2.6 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \end{bmatrix}|$$
$$= 2 - 6.8\alpha + .6|-1 + \alpha| \;=\; 2.6 - 7.4\alpha \quad \text{for } 0 \le \alpha \le 1\,.$$

We see that $\Delta = -7.4$ and

$$\pi(1) = 2.984 > 2.6 + .1\Delta \cdot 1 = 1.86\,.$$

We need to reduce α, and use (4.26) to find[6)]

$$\beta = \frac{7.4}{2(2.984 - 2.6 + 7.4)} = 0.475334 = \alpha \ ,$$

$$\pi(\alpha) = 0.667740 < 2.6 + .1\Delta\alpha \simeq 2.248 \ .$$

Thus, the line search is finished, and the next iterate is

$$\mathbf{x}_{\text{new}} = \mathbf{x} + \alpha\mathbf{h} = [\,0.619733,\ -0.235868\,]^\top \ .$$

To update $\mathbf{W}$ we use (4.28) and find

$$\mathbf{y} = \begin{bmatrix} -0.304214 \\ -2.471737 \end{bmatrix}, \ \dot{y}h \simeq 6.67 > 0 \ , \ \mathbf{W}_{\text{new}} \simeq \begin{bmatrix} 0.943 & -0.044 \\ -0.044 & 2.014 \end{bmatrix} \ .$$

The error estimate computed by (4.29) is $\eta(\mathbf{x}, \boldsymbol{\lambda}) = 0.23$, and the true error[7)] is $\|\mathbf{x}_{\text{new}} - \mathbf{x}^*\|_\infty = 0.26$.

The second model problem is

$$\begin{aligned} \text{minimize} \quad & q(\boldsymbol{\delta}) = 0.440 + 1.239\delta_1 - 0.472\delta_2 \\ & \qquad +0.471\delta_1^2 - 0.044\delta_1\delta_2 + 1.007\delta_1^2 \\ \text{subject to} \quad & d_1(\boldsymbol{\delta}) = -0.380 + 1.239\delta_1 - \delta_2 = 0 \ . \end{aligned}$$

According to Example 3.3 the solution is

$$(\mathbf{h}, \boldsymbol{\lambda}) = \Big([\,0.070550,\ -0.292618\,]^\top,\ 1.064025\Big) \ ,$$

and the penalty function with $\mu = \lambda$ shows that $\alpha = 1$. We get

$$\mathbf{x}_{\text{new}} = [\,0.690283,\ -0.528487\,]^\top$$

$$\eta(\mathbf{x}_{\text{new}}, \boldsymbol{\lambda}) \simeq 0.094 \ , \quad \|\mathbf{x}_{\text{new}} - \mathbf{x}^*\|_\infty \simeq 0.028 \ ,$$

$$\mathbf{W}_{\text{new}} \simeq \begin{bmatrix} 0.908 & 0.250 \\ 0.250 & 2.060 \end{bmatrix} \ .$$

The results from the next iteration steps are

6) The computation in this example was performed with machine accuracy $\varepsilon_{\text{M}} = 5{\cdot}10^{-14}$, but results are shown with at most 6 digits.

7) According to Example 4.3, $\mathbf{x}^* = [\,\sqrt{0.5},\ -0.5\,]^\top$.

k	$\mathbf{x}_k^\top$	λ_k	$\eta(\mathbf{x}_k, \lambda_k)$	$\|\mathbf{x}_k - \mathbf{x}^*\|_\infty$
3	$[0.701733,\ -0.507702]$	1.011291	$7.0{\cdot}10^{-4}$	$7.7{\cdot}10^{-4}$
4	$[0.707111,\ -0.500023]$	1.000498	$5.9{\cdot}10^{-5}$	$2.3{\cdot}10^{-5}$
5	$[0.707107,\ -0.500000]$	0.999990	$8.2{\cdot}10^{-10}$	$4.6{\cdot}10^{-8}$

If we use $\varepsilon = 10^{-8}$ in Algorithm 4.30, we are finished. The errors $\{\|\mathbf{x}_k - \mathbf{x}^*\|_\infty\}$ exhibit superlinear convergence. ∎

Example 4.6. In the table below we give some test results from Powell (1977). The size of each problem is given in the first two columns. The next column gives the number of elements in $\mathcal{B}(\mathbf{x}^*)$ and the number in parenthesis tells, how many of these that are linear. The last two columns give the number of iterations and the number of function calls needed to solve the problem to a desired accuracy of 10^{-5}. For comparison we also give (in parenthesis) the corresponding numbers for the augmented Lagrangian algorithm 4.20.

n	m	$\#\mathcal{B}(\mathbf{x}^*)$		Iterations		Fct. calls	
3	7	1	(1)	5	(4)	7	(30)
5	3	3	(0)	6	(5)	7	(37)
5	15	4	(4)	4	(4)	6	(39)
5	16	5	(3)	2	(5)	3	(64)
15	20	11	(3)	16	(3)	17	(149)

Each function call involves one evaluation of $f(\mathbf{x})$ and $\mathbf{f}'(\mathbf{x})$.

For these examples the Lagrange–Newton method is clearly superior when the number of function evaluations is used as a measure. However, the work load per function evaluation may be much higher for the Lagrange–Newton method since it involves many QP problems. This is especially important when the number of variables and/or constraints is high.

In conclusion, we recommend the Lagrange-Newton method (SQP) when function evaluations are expensive, and the Augmented Lagrangian method when function evaluations are cheap and we have many variables and constraints. ∎

APPENDIX

A. Karush–Kuhn–Tucker Theorem

We shall prove property 2° in Theorem 2.5. Without loss of generality we assume that the active inequality constraints are numbered first:

$$\begin{aligned} \text{Equality constraints}: &\ c_i(\mathbf{x}) = 0,\ i = 1, \ldots, r \\ \text{Active inequality constraints}: &\ c_i(\mathbf{x}) = 0,\ i = r{+}1, \ldots, p \\ \text{Inactive constraints}: &\ c_i(\mathbf{x}) > 0,\ i = p{+}1, \ldots, m\ . \end{aligned}$$

The comments on 3° in the theorem and the definition of the Lagrange function imply that 1° has the form

$$\mathbf{f}\,'(\mathbf{x}) = \sum_{i=1}^{p} \lambda_i \mathbf{a}_i \quad \text{with } \mathbf{a}_i = \mathbf{c}_i'(\mathbf{x})\ . \tag{A.1}$$

We shall prove that if one (or more) of the $\{\lambda_i\}_{i=r+1}^{p}$ is negative, then $\mathbf{x}$ is **not** a local, constrained minimizer:

For the sake of simplicity, assume that the gradients $\{\mathbf{a}_i\}_{i=1}^{p}$ are linearly independent and that $\lambda_p < 0$. Then we can decompose $\mathbf{a}_p$ into $\mathbf{v}$, its orthogonal projection on the subspace spanned by $\{\mathbf{a}_i\}_{i=1}^{p-1}$ and $\mathbf{h}$, which is orthogonal to this subspace,

$$\mathbf{a}_p = \mathbf{v} + \mathbf{h} \quad \text{with} \quad \mathbf{a_i}^\top \mathbf{h} = 0 \ \text{ for } \ i = 1, \ldots, p{-}1\ .$$

For small values of $\|\alpha\mathbf{h}\|$ we use the Taylor series (1.7) for the constraint functions to see that

$$c_i(\mathbf{x}{+}\alpha\mathbf{h}) \ \simeq\ c_i(\mathbf{x}){+}\alpha\mathbf{h}^\top\mathbf{a_i} \ = \begin{cases} 0 & \text{for } \ i = 1, \ldots, p{-}1 \\ \alpha\mathbf{h}^\top\mathbf{h} & \text{for } \ i = p \end{cases}\ .$$

This shows, that for $\alpha > 0$ and sufficiently small, $\mathbf{x}{+}\alpha\mathbf{h}$ is feasible. Further, from the Taylor series (1.6) for the cost function and (A.1) we get

$$\begin{aligned} f(\mathbf{x} + \alpha\mathbf{h}) \ &\simeq\ f(x) + \alpha\mathbf{h}^\top \mathbf{f}\,'(\mathbf{x}) \\ &=\ f(x) + \alpha\mathbf{h}^\top \left(\textstyle\sum_{i=1}^{p} \lambda_i \mathbf{a}_i\right) \\ &=\ f(x) + \alpha\lambda_p \mathbf{h}^\top\mathbf{h}\ , \end{aligned}$$

showing that $f(\mathbf{x}{+}\alpha\mathbf{h}) < f(\mathbf{x})$ for $\alpha > 0$, since $\lambda_p < 0$.

Thus, we have shown that at a local, constrained minimizer all the Lagrange multipliers for inequality constraints are nonnegative. □

REFERENCES

1. C.G. Broyden (1967): *Quasi–Newton methods and their Application to Function Minimization*, Math. Comps., **21**, 368–381.
2. O. Caprani, K. Madsen and H.B. Nielsen (2002): *Interval Analysis*, IMM, DTU. Available as http://www.imm.dtu.dk/courses/02611/IA.pdf
3. I.S. Duff (1993): *The Solution of Augmented Systems*, Report RAL-93-084, Rutherford Appleton Laboratory.
4. A.V. Fiarco and G.P. McCormick (1968): *Nonlinear Programming*, Wiley.
5. R. Fletcher (1993): *Practical Methods of Optimization*, Wiley.
6. P.E. Frandsen, K. Jonasson, H.B. Nielsen and O. Tingleff (2004): *Unconstrained Optimization*, IMM, DTU. Available as http://www.imm.dtu.dk/courses/02611/uncon.pdf
7. P.E. Gill and W. Murray (1974): *Newton type methods for Linearly Constrained Optimization* in P.E. Gill and W. Murray (eds): "Numerical Methods for Constrained Optimization", Academic Press.
8. G. Golub and C.F. Van Loan (1996): *Matrix Computations*, 3rd edition. John Hopkins Press.
9. S.P. Han (1976): *Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems*, Math. Progr. **11**, 263–282.

10. G.H. Kuhn and A.W. Tucker (1951): *Nonlinear Programming* in J. Neyman (ed) "Procedings of the 2nd Berkeley Symposion on Mathematical Statistics and Probability", pp 481–493, University of California Press.
11. K. Madsen (1995): *Optimering under bibetingelser* (in Danish), IMM, DTU.
12. K. Madsen and H.B. Nielsen (2002): *Supplementary Notes for 02611 Optimization and Data Fitting*, IMM, DTU. Available as http://www.imm.dtu.dk/courses/02611/SN.pdf
13. H.B. Nielsen (1999): *Algorithms for Linear Optimization*, IMM, DTU. Available as http://www.imm.dtu.dk/courses/02611/ALO.pdf
14. M.J.D. Powell (1969): *A Method for Nonlinear Constraints in Minimization Problems*, in R. Fletcher (ed): "Optimization", Academic Press.
15. M.J.D. Powell (1977): *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, in "Numerical Analysis, Dundee 1977", Lecture Notes in Mathematics **630**, Springer.

INDEX