3 Constrained Optimization

In this section we will see the constrained optimization theory. The theory is a bit deep and we will see only a few of the main results in this course. A general formulation of constrained optimization problems has the form

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \quad \begin{cases} c_j^{\text{eq}}(\mathbf{x}) = 0 & j = 1, ..., m^{\text{eq}} \\ c_l^{\text{ieq}}(\mathbf{x}) \leq 0 & l = 1, ..., m^{\text{ieq}} \end{cases}, \tag{13}$$

where f, c_j^{eq}, and c_j^{ieq} are all smooth functions. $f(\mathbf{x})$, as before, is the objective that we wish to minimize. $c_j^{\text{eq}}(\mathbf{x})$ are *equality* constraints, and $c_j^{\text{ieq}}(\mathbf{x})$ are *inequality* constraints. We define a *feasible set* to be the set of all points that satisfy the constraints:

$$\Omega = \left\{\mathbf{x} | c_j^{\text{eq}}(\mathbf{x}) = 0, \quad j = 1, ..., m^{\text{eq}}; \quad c_l^{\text{ieq}}(\mathbf{x}) \leq 0, \quad l = 1, ..., m^{\text{ieq}}\right\}.$$

We can write (13) compactly as $\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$. The focus in this section is to characterize the solutions of (13). Recall that for unconstrained minimization we had the following sufficient optimality condition: *Any point $\mathbf{x}^*$ at which $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive definite is a strong local minimizer of f.* Our goal now is to define similar optimality conditions to constrained optimization problems.

We have seen already that global solutions are difficult to find even when there are no constraints. The situation may be improved when we add constraints, since the feasible set might exclude many of the local minima, and it may be easy to pick the global minimum from those that remain. However, constraints can also make things much more difficult. As an example, consider the problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{x}\|_2 \quad \text{s.t.} \quad \|\mathbf{x}\|_2^2 \geq 1.$$

Without the constraint, this is a convex quadratic problem with unique minimizer $\mathbf{x} = 0$. When the constraint is added, any vector $\mathbf{x}$ with $\|\mathbf{x}\|_2 = 1$ solves the problem. There are infinitely many such vectors (hence, infinitely many local minima) whenever $n \geq 2$.

Definitions of the different types of local solutions are simple extensions of the corresponding definitions for the unconstrained case, except that now we restrict consideration to the feasible points in the neighborhood of $\mathbf{x}^*$. We have the following definition.

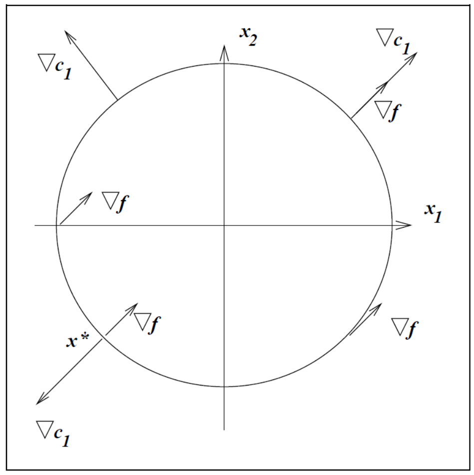

Figure 8: The single equality constraint example.

Definition 3 (Local solution of the constrained problem)**.** *A vector* $\mathbf{x}^*$ *is a local solution of the problem* (13) *if* $\mathbf{x}^* \in \Omega$ *(*$\mathbf{x}^*$ *satisfies the constraints) and there is a neighborhood* $\mathcal{N}$ *of* $\mathbf{x}^*$ *such that* $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ *for* $\mathbf{x} \in \mathcal{N} \cap \Omega$*.*

3.1 Equality-constrained optimization – Lagrange multipliers

To introduce the basic principles behind the characterization of solutions of general constrained optimization problems, we will first consider only equality constraints. We start with a relatively simple case of constrained optimization with a single equality constraint.

Our first example is a two-variable problem with a single equality constraint.

Example 8 (A single equality constraint)**.** *Consider the following problem:*

$$\min_{\mathbf{x} \in \mathbb{R}^n} x_1 + x_2 \quad \textit{subject to} \quad x_1^2 + x_2^2 - 2 = 0, \tag{14}$$

In the language of Equation (13) *we have* $m^{\text{eq}} = 1$ *and* $m^{\text{ieq}} = 0$*, with* $c_1^{\text{eq}}(\mathbf{x}) = x_1^2 + x_2^2 - 2$*.*

We can see by inspecting Fig 8 that the feasible set for this problem is the circle of radius $\sqrt{2}$ *centered at the origin—–just the boundary of this circle, not its interior. The solution* $\mathbf{x}^*$ *is obviously* $[-1, -1]^\top$ *. From any other point on the circle, it is easy to find a way to move that stays feasible (that is, remains on the circle) while decreasing* f*.*

We also see that at the optimal point $\mathbf{x}^*$*, the objective gradient* ∇f *is parallel to the constraint gradient* $\nabla c_1^{\text{eq}}(\mathbf{x}^*)$*. In other words, there exists a scalar* λ_1 *such that*

$$\nabla f + \lambda_1 \nabla c_1^{\text{eq}}(\mathbf{x}^*) = 0, \tag{15}$$

and in this particular case where $\nabla f = [1,1]^\top$*, and* $\nabla c_1^{\text{eq}} = [2x_1, 2x_2]^\top$*, we have* $\lambda_1 = \frac{1}{2}$*, for* $\mathbf{x} = [-1,-1]^\top$*.*

We can reach the same conclusion using the Taylor expansion. Assume a feasible point $\mathbf{x}$*. To retain feasibility, any small movement in the direction* $\mathbf{d}$ *from* $\mathbf{x}$ *has to satisfy the constraint*

$$0 = c_1^{\text{eq}}(\mathbf{x}+\mathbf{d}) \approx c_1^{\text{eq}}(\mathbf{x}) + \langle \nabla c_1^{\text{eq}}(\mathbf{x}), \mathbf{d} \rangle = \langle \nabla c_1^{\text{eq}}(\mathbf{x}), \mathbf{d} \rangle.$$

So any movement at the direction $\mathbf{d}$ *from the point* $\mathbf{x}$ *has to satisfy*

$$\boxed{\langle \nabla c_1^{\text{eq}}(\mathbf{x}), \mathbf{d} \rangle = 0.} \tag{16}$$

On the other hand, we have seen earlier than any descent direction has to satisfy

$$\boxed{\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle < 0,} \tag{17}$$

because $f(\mathbf{x}+\mathbf{d}) - f(\mathbf{x}) \approx \langle \nabla f(\mathbf{x}), \mathbf{d} \rangle$*.*

If there exists a direction $\mathbf{d}$*, satisfying* (16)-(17)*, then we can say that improvement in* f *is possible under the constraints. It follows that a necessary condition for optimality for our problem is that there is not a direction* $\mathbf{d}$ *satisfying both* (16)-(17)*. The only way that such a direction cannot exist is if* $\nabla f(\mathbf{x})$ *and* $\nabla c_1^{\text{eq}}(\mathbf{x})$ *are parallel, otherwise*

$$\mathbf{d} = \left(I - \frac{1}{\nabla c_1^{\text{eq}}(\mathbf{x})^\top \nabla c_1^{\text{eq}}(\mathbf{x})} \nabla c_1^{\text{eq}}(\mathbf{x}) \nabla c_1^{\text{eq}}(\mathbf{x})^\top \right) \nabla f(\mathbf{x}), \tag{18}$$

is a descent direction. It is the projection of $\nabla f(\mathbf{x})$ *to be orthogonal to* ∇c_1^{eq}*, and this direction satisfies the constraints at least at first order. To eliminate this option, we will demand that there is no such direction and zero* (18)*:*

$$\frac{1}{\nabla c_1^{\text{eq}}(\mathbf{x})^\top \nabla c_1^{\text{eq}}(\mathbf{x})} \nabla c_1^{\text{eq}}(\mathbf{x}) \nabla c_1^{\text{eq}}(\mathbf{x})^\top \nabla f(\mathbf{x}) = \nabla f(\mathbf{x}).$$

This means that ∇f *and* ∇c_1^{eq} *are parallel.*

From another point of view: We can look at the right hand side of Eq. (18) *as a least squares solution of*

$$\lambda_1 = \arg\min_{\lambda} \|\nabla c_1 \lambda + \nabla f\|_2^2.$$

That is because its solution is: $\lambda_1 = -\frac{\nabla c_1^{\mathsf{eq}}(\mathbf{x})^\top \nabla f(\mathbf{x})}{\nabla c_1^{\mathsf{eq}}(\mathbf{x})^\top \nabla c_1^{\mathsf{eq}}(\mathbf{x})}$, *and we get to* (18) *inside the* ℓ_2 *norm. We conclude that the least squares minimization above must be satisfied with a value of 0. It means that there exists a* λ_1 *satisfying* (15).

Because of the reasoning above, we can "pack" the objective and its equality constraints into one function called the Lagrangian *function*

$$\mathcal{L}(\mathbf{x}, \lambda_1) = f(\mathbf{x}) + \lambda_1 c_1^{\mathsf{eq}}(\mathbf{x}),$$

for which $\nabla_{\mathbf{x}} \mathcal{L} = \nabla_{\mathbf{x}} f(\mathbf{x}) + \lambda_1 \nabla_{\mathbf{x}} c_1^{\mathsf{eq}}(\mathbf{x})$. *Hence, at the solution* $\mathbf{x}^*$ *there exists a scalar* λ_1 *such that*

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda_1^*) = 0.$$

Note that the condition $\nabla_{\lambda_1} \mathcal{L} = 0$ *provides the constraint* $c_1^{\mathsf{eq}}(\mathbf{x}) = 0$. *This observation suggests that we can search for solutions of the equality-constrained problem by searching for stationary points of the Lagrangian function. The scalar quantity* λ_1 *is called a Lagrange multiplier for the constraint* $c_1^{\mathsf{eq}}(\mathbf{x}) = 0$. *The condition above appears to be necessary for an optimal solution of the problem, but it is clearly not sufficient. For example, the point* $[1, 1]^\top$ *also satisfies it in our example, but it is the maximum of the function under the constraint.*

Back to our general equality constrained problem, which we write as

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{c}^{\mathsf{eq}}(\mathbf{x}) = 0, \tag{19}$$

where $\mathbf{c}^{\mathsf{eq}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{m^{\mathsf{eq}}}$ is the constraints vector function, that is

$$\mathbf{c}^{\mathsf{eq}}(\mathbf{x}) = \begin{bmatrix} c_1^{\mathsf{eq}}(\mathbf{x}) \\ c_2^{\mathsf{eq}}(\mathbf{x}) \\ \vdots \\ c_{m^{\mathsf{eq}}}^{\mathsf{eq}}(\mathbf{x}) \end{bmatrix} = 0.$$

To obtain the optimality conditions we will first make an assumption on a point $\mathbf{x}^*$ regarding the constraints.

Definition 4 (LICQ for equality constrained optimization)**.** *Given the point* $\mathbf{x}^*$*, we say that the linear independence constraint qualification (LICQ) holds if the set of constraint gradients* ∇c_i^{eq} *is linearly independent, or equivalently, if the Jacobian of the constraints vector* $\mathbf{J}^{\mathsf{eq}}(\mathbf{x}^*)$ *is a full-rank matrix.*

Note that if this condition holds, none of the constraint gradients can be zero. This assumption comes to simplify our derivations and is not really necessary from a practical point of view. For example, it breaks if we write one of the constraints twice. This clearly should have any practical influence on our ability to solve problems.

A generalization of our conclusion from the previous example is as follows. In order for $\mathbf{x}^*$ to be a stationary point, there shouldn't be a direction $\mathbf{d}$ such that $\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle < 0$ and $\langle \nabla c_i^{\mathsf{eq}}(\mathbf{x}), \mathbf{d} \rangle = 0$ for $i = 1, ..., m^{\mathsf{eq}}$. In other words, the gradient cannot have a component that is orthogonal to all the constraints' gradients. More explicitly let

$$\mathbf{J}^{\mathsf{eq}} = \begin{bmatrix} - & \nabla c_1^{\mathsf{eq}}(\mathbf{x}) & - \\ - & \nabla c_2^{\mathsf{eq}}(\mathbf{x}) & - \\ & \vdots & \\ - & \nabla c_{m^{\mathsf{eq}}}^{\mathsf{eq}}(\mathbf{x}) & - \end{bmatrix}.$$

be the Jacobian of the constraints, the demand above will happen only if (similarly to the 1D case) the orthogonal projection of ∇f on the span of the rows of the Jacobian is zero (dropping the $^{\mathsf{eq}}$ notation):

$$\left(I - \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}\mathbf{J}\right)\nabla f = 0 \Rightarrow \min_{\boldsymbol{\lambda}} \|\mathbf{J}^T\boldsymbol{\lambda} + \nabla f\|_2^2 = 0,$$

since the minimizer of the LS problem is $\boldsymbol{\lambda} = (\mathbf{J}\mathbf{J}^T)^{-1}\mathbf{J}\nabla f$. It means that if we project ∇f to be orthogonal to all the rows of $\mathbf{J}$, we end up with zero.

This also means that $\nabla f(\mathbf{x})$ is a linear combination of $\mathbf{J}^{\mathsf{eq}}$'s rows, and we get the condition:

$$\nabla f(\mathbf{x}) + \sum_{i=1}^{m^{\mathsf{eq}}} \lambda_i \nabla c_i^{\mathsf{eq}}(\mathbf{x}) = \nabla f(\mathbf{x}) + (\mathbf{J}^{\mathsf{eq}})^\top \boldsymbol{\lambda} = 0.$$

Consequently, given an equality constrained problem (19), we write its Lagrangian function as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{c}^{\mathsf{eq}}(\mathbf{x}),$$

where the vector $\boldsymbol{\lambda} \in \mathbb{R}^{m^{\mathsf{eq}}}$ is the Lagrange multipliers vector.

Theorem 2 (Lagrange multipliers for equality constrained minimization)**.** *Let $\mathbf{x}^*$ be a local solution of* (19)*, in particular satisfying the equality constraints $\mathbf{c}^{\mathsf{eq}}(\mathbf{x}^*) = 0$, and the LICQ condition. Then there exists a unique vector $\boldsymbol{\lambda}^* \in \mathbb{R}^{m^{\mathsf{eq}}}$ called the Lagrange multipliers vector which satisfies*

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \nabla f(\mathbf{x}^*) + (\mathbf{J}^{\mathsf{eq}})^\top \boldsymbol{\lambda}^* = 0.$$

We will not prove this theorem, but note that $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0 \Rightarrow \mathbf{c}^{\mathsf{eq}}(\mathbf{x}^*) = 0$, which allows us to formulate a method for solving equality constrained optimization.

Back to our example of a single equality constraint

$$\min_{\mathbf{x} \in \mathbb{R}^n} x_1 + x_2 \quad \text{subject to} \quad x_1^2 + x_2^2 - 2 = 0. \tag{20}$$

The Lagrangian of this method is $\mathcal{L}(\mathbf{x}, \lambda_1) = x_1 + x_2 + \lambda_1(x_1^2 + x_2^2 - 2)$ and the solution of this problem is given by

$$\nabla \mathcal{L} = 0 \Rightarrow \begin{cases} 1 + 2x_1\lambda_1 = 0 \\ 1 + 2x_2\lambda_1 = 0 \\ x_1^2 + x_2^2 - 2 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -\frac{1}{2\lambda_1} \\ x_2 = -\frac{1}{2\lambda_1} \\ \frac{1}{4\lambda_1^2} + \frac{1}{4\lambda_1^2} = 2 \end{cases} \Rightarrow \begin{cases} \lambda_1 = \pm\frac{1}{2} \\ x_1 = \mp 1 \\ x_2 = \mp 1 \end{cases},$$

As discussed before, we got two stationary points for the Lagrangian. How do we decide which one of them is a local minimum? The following theorem says that the Hessian of $\mathcal{L}$ with respect to $\mathbf{x}$ has to be positive definite with respect to the directions that satisfy the constraints (or, the directions which are orthogonal to the constraints gradients.).

Theorem 3 (2^{nd} order necessary conditions for equality constrained minimization)**.** *Let $\mathbf{x}^*$ be a minimum solution and let $\boldsymbol{\lambda}^*$ be a corresponding Lagrange multiplier vector satisfying $\nabla \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0$. Also assume that the LICQ condition is satisfied. Then the following must hold:*

$$\mathbf{y}^\top \nabla^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*)\mathbf{y} \geq 0 \quad \forall \quad \mathbf{y} \in \mathbb{R}^n \quad s.t. \quad \mathbf{J}^{\mathsf{eq}}\mathbf{y} = 0.$$

In our example, $\mathcal{L}(\mathbf{x}, \lambda_1) = x_1 + x_2 + \lambda_1(x_1^2 + x_2^2 - 2)$, and

$$\nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \begin{bmatrix} 2\lambda_1 & 0 \\ 0 & 2\lambda_1 \end{bmatrix} \quad \nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^* = \pm 1/2) = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix}$$

It follows that the minimum is obtained at $\lambda_1^* = 1/2$ where $\nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = I \succ 0$. This matrix is positive definite with respect to any vector, and in particular to those vectors that are orthogonal to the constraint gradient. In the other point ($\lambda_1 = -1/2$), we encounter the case of the negative definite Hessian, and hence we have a local maximum.

3.2 The KKT optimality conditions for general constrained optimization

Suppose now that we are considering the general case of constrained optimization, allowing inequality constraints as well. Recall the following problem definition:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \quad \begin{cases} c_j^{\mathsf{eq}}(\mathbf{x}) = 0 & j = 1, ..., m^{\mathsf{eq}} \\ c_l^{\mathsf{ieq}}(\mathbf{x}) \le 0 & l = 1, ..., m^{\mathsf{ieq}} \end{cases} . \tag{21}$$

It turns out that for this problem we should make a distinction between *active* inequality constraints and *inactive* inequality constraints. Assume a local solution $\mathbf{x}^*$. The active constraints are those constraints for which $c_l^{\mathsf{ieq}}(\mathbf{x}^*) = 0$ and the inactive constraints are those constraints for which $c_l^{\mathsf{ieq}}(\mathbf{x}^*) < 0$.

Example 9. *Consider the following problem which is similar to the previous one, only now we have an inequality constraint:*

$$\min_{\mathbf{x}\in\mathbb{R}^n} x_1 + x_2 \quad \textit{subject to} \quad x_1^2 + x_2^2 - 2 \le 0. \tag{22}$$

Similarly to the previous problem, the solution of this problem lies at the point $[-1, -1]^\top$, where the constraint is active. The Lagrangian is again $\mathcal{L}(\mathbf{x}, \lambda_1) = x_1 + x_2 + \lambda_1(x_1^2 + x_2^2 - 2)$, and the minimum is obtained at the point satisfying $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \lambda_1^*) = 0$. But now, it turns out that the sign of the Lagrange multiplier $\lambda_1^* = \frac{1}{2}$ has a significant role.

Similarly to before, we need that any search direction $\mathbf{d}$ *will satisfy*

$$c_1^{\text{ieq}}(\mathbf{x}+\mathbf{d}) \approx c_1^{\text{ieq}}(\mathbf{x}) + \langle \nabla c_1^{\text{ieq}}(\mathbf{x}), \mathbf{d} \rangle \leq 0.$$

If the constraint c_1^{ieq} *is inactive at the point* $\mathbf{x}$ *we can move at any direction, and in particular we can move at the descent direction* $\nabla f(\mathbf{x})$ *and decrease the objective. However, if the constraint is active, then* $c_1^{\text{ieq}}(\mathbf{x}) = 0$*, and to satisfy the constraint we have the condition*

$$\langle \nabla c_1^{\text{ieq}}(\mathbf{x}), \mathbf{d} \rangle \leq 0. \tag{23}$$

In order for $\mathbf{d}$ *to be a legal descent direction it needs to satisfy both* $\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle < 0$ *and* (23)*. If we wish that* $\mathbf{d}$ *is not a descent direction, we need the inner products not to be both negative. For this we must require that* $-\nabla f$ *and* ∇c_1 *will be at the same direction. That is:*

$$\nabla f + \lambda_1 \nabla c_1^{\text{ieq}}(\mathbf{x}^*) = 0,$$

with $\lambda_1 \geq 0$*.*

From the example above we learn that **active** inequality constraints act as equality constraints but require the lagrange multiplier to be **positive** at stationary points. If we have multiple inequality constraints, $\{c_l^{\text{ieq}}(\mathbf{x}) \leq 0\}_{l=1}^{m_{\text{eq}}}$, then we require that for a descent direction $\mathbf{d}$, each of them **which is active** will satisfy $\langle \nabla c_l^{\text{ieq}}(\mathbf{x}), \mathbf{d} \rangle \leq 0$, but if

$$\nabla f + \sum_{\imath \text{ active}} \lambda_l \nabla c_l^{\text{ieq}}(\mathbf{x}^*) = 0,$$

with $\lambda_l \geq 0$, then

$$\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle + \sum_{\imath \text{ active}} \lambda_l \langle \nabla c_l^{\text{ieq}}(\mathbf{x}^*), \mathbf{d} \rangle = 0,$$

and $\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle \geq 0$ must hold. This means that $\mathbf{d}$ cannot be a descent direction under these conditions.

We will now state the necessary conditions for a solution of a general constrained minimization problem.

Definition 5 (Active set)**.** *The active set* $\mathcal{A}(\mathbf{x})$ *at any feasible* $\mathbf{x}$ *is the union of inequality constrains which are satisfied with exact equality* $\{l : c_l^{\text{ieq}}(\mathbf{x}) = 0\}$*.*

We define the Lagrangian of the problem (21) by

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^{\mathsf{eq}}, \boldsymbol{\lambda}^{\mathsf{ieq}}) = f(\mathbf{x}) + (\boldsymbol{\lambda}^{\mathsf{eq}})^\top \mathbf{c}^{\mathsf{eq}}(\mathbf{x}) + (\boldsymbol{\lambda}^{\mathsf{ieq}})^\top \mathbf{c}^{\mathsf{ieq}}(\mathbf{x}),$$

where $\boldsymbol{\lambda}^{\mathsf{eq}} \in \mathbb{R}^{m^{\mathsf{eq}}}$ and $\boldsymbol{\lambda}^{\mathsf{ieq}} \in \mathbb{R}^{m^{\mathsf{ieq}}}$ are the Lagrange multiplier vectors for the equality and inequality constraints respectively. The next theorem states the first order necessary conditions, known as the Karush—Kuhn—Tucker (KKT) conditions.

Theorem 4 (First-Order Necessary Conditions (KKT))**.** *Suppose that* $\mathbf{x}^*$ *is a local solution of* (21)*, and that the LICQ holds at* $\mathbf{x}^*$*. Then there are Lagrange multiplier vectors* $\boldsymbol{\lambda}^{\mathsf{eq}} \in \mathbb{R}^{m^{\mathsf{eq}}}$ *and* $\boldsymbol{\lambda}^{\mathsf{ieq}} \in \mathbb{R}^{m^{\mathsf{ieq}}}$*, such that the following hold:*

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^{\mathsf{eq}*}, \boldsymbol{\lambda}^{\mathsf{ieq}*}) = \nabla f(\mathbf{x}^*) + (\mathbf{J}^{\mathsf{eq}})^\top \boldsymbol{\lambda}^{\mathsf{eq}*} + (\mathbf{J}^{\mathsf{ieq}})^\top \boldsymbol{\lambda}^{\mathsf{ieq}*} = 0 \tag{24}$$

$$\mathbf{c}^{\mathsf{eq}}(\mathbf{x}^*) = 0 \tag{25}$$

$$\mathbf{c}^{\mathsf{ieq}}(\mathbf{x}^*) \leq 0 \tag{26}$$

$$\boldsymbol{\lambda}^{\mathsf{ieq}*} \geq 0 \tag{27}$$

$$(\textit{Complementary slackness}) \quad \textit{for} \;\; l = 1, ..., m^{\mathsf{ieq}} \;\; \lambda_l^{\mathsf{ieq}*} c_l^{\mathsf{ieq}}(\mathbf{x}) = 0 \tag{28}$$

The conditions (26)-(28) *may be replaced with* $\lambda_l^{\mathsf{ieq}*} > 0$ *for the active constraints* $l \in \mathcal{A}(\mathbf{x}^*)$ *and* $\lambda_l^{\mathsf{ieq}*} = 0$ *for the inactive ones.*

To know whether a stationary point is a minimum or a maximum we have the following theorem:

Theorem 5 (2^{nd} order necessary conditions for general constrained minimization)**.** *Let* $\mathbf{x}^*$ *be a minimum solution satisfying the KKT conditions and let* $\boldsymbol{\lambda}^{\mathsf{eq}*}$, $\boldsymbol{\lambda}^{\mathsf{ieq}*}$ *be the corresponding Lagrange multiplier vectors. Also assume that the LICQ condition is satisfied. Then the following must hold:*

$$\mathbf{y}^\top \nabla^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^{\mathsf{eq}*}, \boldsymbol{\lambda}^{\mathsf{ieq}*}) \mathbf{y} \geq 0 \quad \forall \quad \mathbf{y} \in \mathbb{R}^n \quad s.t. \quad \begin{cases} \mathbf{J}^{\mathsf{eq}} \mathbf{y} = 0 \\ \nabla c_l^{\mathsf{ieq}}(\mathbf{x}^*))^\top \mathbf{y} = 0 \quad l \in \mathcal{A}(\mathbf{x}^*) \end{cases}$$

Example 10. *Consider the following problem*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left(x_1 - \frac{3}{2}\right)^2 + \left(x_2 - \frac{1}{8}\right)^4 \quad s.t. \quad \begin{cases} x_1 + x_2 - 1 \leq 0 \\ x_1 - x_2 - 1 \leq 0 \\ -x_1 + x_2 - 1 \leq 0 \\ -x_1 - x_2 - 1 \leq 0 \end{cases}. \tag{29}$$

This time, we have four inequality constraints. In principle, we should try every combination of active and inactive constraints and see if the resulting $\mathbf{x}^*$ *and* $\boldsymbol{\lambda}^*$ *that are achieved by* $\nabla_{\mathbf{x}}\mathcal{L} = 0$ *satisfy the KKT conditions. It turns out that the solution here is* $\mathbf{x}^* = [1, 0]^\top$, *and that the first and second constraints are active at this point. Denoting them by* c_1 *and* c_2 *we have*

$$\nabla f(\mathbf{x}^*) = \begin{bmatrix} -1 \\ -\frac{1}{128} \end{bmatrix} \quad \nabla c_1(\mathbf{x}^*) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \nabla c_2(\mathbf{x}^*) = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

and the vector $\boldsymbol{\lambda} = [\frac{129}{256}, \frac{127}{256}, 0, 0]^\top$.

3.3 Penalty and Barrier methods

In the previous section we saw that in order to solve a constrained optimization using the KKT conditions, we may need to solve several large linear systems, each time checking a different combination of active and inactive inequality constraints. This is a option for solving the problem but it does not align with the optimization methods we've seen so far. In this section, we will see two closely related approaches for solving constrained optimization—the penalty and barrier approaches. Both of these approach convert the constrained problem into a somewhat equivalent unconstrained problem which can be solved by SD, Newton, or any other method for unconstrained optimization.

Penalty methods. We again assume that we have the general constrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \quad \begin{cases} c_j^{\text{eq}}(\mathbf{x}) = 0 & j = 1, ..., m^{\text{eq}} \\ c_l^{\text{ieq}}(\mathbf{x}) \leq 0 & l = 1, ..., m^{\text{ieq}} \end{cases}, \tag{30}$$

and now we wish to transform it to an equivalent unconstrained problem. In the penalty approach, we rewrite the problem as

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) + \mu\left(\sum_{j=1}^{m^{\mathsf{eq}}}\rho_j(c_j^{\mathsf{eq}}(\mathbf{x})) + \sum_{l=1}^{m^{\mathsf{ieq}}}\rho_l(\max\{0, c_l^{\mathsf{ieq}}(\mathbf{x})\})\right) \tag{31}$$

where $\mu > 0$ is a balancing penalty parameter, and $\{\rho_j(x)\}$ and $\{\rho_l(x)\}$ are scalar penalty functions which are bounded from below and get their minimum at 0 (it is preferable that these are non-negative but this is not mandatory). In addition, these functions should monotonically increase as we move away from zero. The common choice for a penalty is the quadratic function $\rho(x) = x^2$, which has a minimum at 0. Using this choice, if we set $\mu \to \infty$ then the solution of (31) will be equal to that of (30). We will focus on this choice in this course.

The problem (31) is an unconstrained optimization problem which can be solved using the methods we've learned to far. For $\rho(x) = x^2$, however, it turns out that the problem becomes more and more ill-conditioned as $\mu \to \infty$, which makes standard first-order approaches like SD and CD slow. Therefore, in practice we will have a continuation strategy where we solve the problem for iteratively increasing values of the penalty μ. That is we will define

$$\mu_0 < \mu_1 < ... < \infty$$

and iteratively solve the problem for each of those $\mu's$, each time using the previous solution as an initial guess for the next problem. We will stop when μ is large enough such that the constraints are *reasonably* fulfilled. That is, for our largest value of μ, it may be that for example one of the equality constraints satisfies $c(\mathbf{x}^*) = \varepsilon$. Whether ε is small enough or not will depend on the application that yielded the optimization problem.

Remark 1. *Another popular choice for a penalty is the exact penalty function $\rho(x) = |x|$ (absolute value). It is popular because unlike the quadratic function, the constraints will be exactly fulfilled for some moderate μ and not only for $\mu \to \infty$. The downside here is that this function is non-smooth and significantly complicates the optimization process. We will not consider this choice further in this chapter.*

Example 11. *Consider again the following problem*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left(x_1 - \frac{3}{2}\right)^2 + \left(x_2 - \frac{1}{8}\right)^4 \quad s.t. \quad \begin{cases} x_1 + x_2 - 1 \leq 0 \\ x_1 - x_2 - 1 \leq 0 \\ -x_1 + x_2 - 1 \leq 0 \\ -x_1 - x_2 - 1 \leq 0 \end{cases} . \tag{32}$$

Previously, we had four inequality constraints, and only the first two where active at the solution. Using the Lagrange multipliers approach we had to check several combinations of active/inactive constraints. We will now use the penalty version of this approach using $\rho = x^2$, *which is vector form becomes*

$$\min_{\mathbf{x}\in\mathbb{R}^n} f_\mu(\mathbf{x}) = \left(x_1 - \frac{3}{2}\right)^2 + \left(x_2 - \frac{1}{8}\right)^4 + \frac{\mu}{2}\left\|\max\{A\mathbf{x} - \mathbf{1}, \mathbf{0}\}\right\|_2^2, \tag{33}$$

where the matrix A *is*

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}.$$

We will solve this problem using Steepest Descent (SD), and the gradient for SD is given by

$$\nabla f_\mu = \begin{bmatrix} 2(x_1 - \frac{3}{2}) \\ 4(x_2 - \frac{1}{8})^3 \end{bmatrix} + \mu A^\top (\mathbf{I}_{A\mathbf{x}-\mathbf{1}>\mathbf{0}} \odot (A\mathbf{x} - \mathbf{1})),$$

where $\odot$ *is the Hadamard point-wise vector product (same as the operator* `.*` *is Julia/Matlab), and* $\mathbf{I}_{A\mathbf{x}-\mathbf{1}>\mathbf{0}}$ *is an indicator vector*

$$(\mathbf{I}_{A\mathbf{x}-\mathbf{1}>\mathbf{0}})_i = \begin{cases} 1 & \textit{if } (A\mathbf{x})_i - 1 > \mathbf{0} \\ 0 & \textit{otherwise} \end{cases} .$$

```
using PyPlot;
close("all");
xx = -0.5:0.01:1.5
yy= -1.0:0.01:1.0;
m = length(xx);
X = repmat(xx ,1,length(yy))';
Y = repmat(yy ,1,length(xx));

oVec = ones(4);
A = [1.0 1 ; 1 -1 ; -1 1 ; -1 -1];
f = (x,mu)->(x[1]-3/2)^2 + (x[2]-1/8)^4 + mu*norm(max(A*x - oVec,0.0)).^2;
g = (x,mu)->[2*(x[1]-3/2) ; 4*(x[2]-1/8).^3] + mu*2*A'*(((A*x- oVec).>0.0).*(A*x - oVec));

F = (X-3/2).^2 + (Y-1/8).^4
C = (max(X + Y - 1,0).^2 + max(-X + Y - 1,0).^2 + max(-X - Y - 1,0).^2 + max(X - Y - 1,0).^2);

figure();
mu = 0.0;
Fc = F + mu*C;
subplot(3,2,1); contour(X,Y,Fc,200); #hold on; axis image;
xlabel("x"); ylabel("y"); title(string("Penalty for mu = ",mu))

mu = 0.01;
Fc = F + mu*C;
subplot(3,2,2); contour(X,Y,Fc,200); #hold on; axis image;
xlabel("x"); ylabel("y"); title(string("Penalty for mu = ",mu))

mu = 0.1;
Fc = F + mu*C;
subplot(3,2,3); contour(X,Y,Fc,200); #hold on; axis image;
xlabel("x"); ylabel("y"); title(string("Penalty for mu = ",mu))

mu = 1;
Fc = F + mu*C;
subplot(3,2,4); contour(X,Y,Fc,200); #hold on; axis image;
xlabel("x"); ylabel("y"); title(string("Penalty for mu = ",mu))

mu = 10;
Fc = F + mu*C;
subplot(3,2,5); contour(X,Y,Fc,500); #hold on; axis image;
xlabel("x"); ylabel("y"); title(string("Penalty for mu = ",mu))

mu = 100;
Fc = F + mu*C;
subplot(3,2,6); contour(X,Y,Fc,1000); #hold on; axis image;
xlabel("x"); ylabel("y"); title(string("Penalty for mu = ",mu))
```

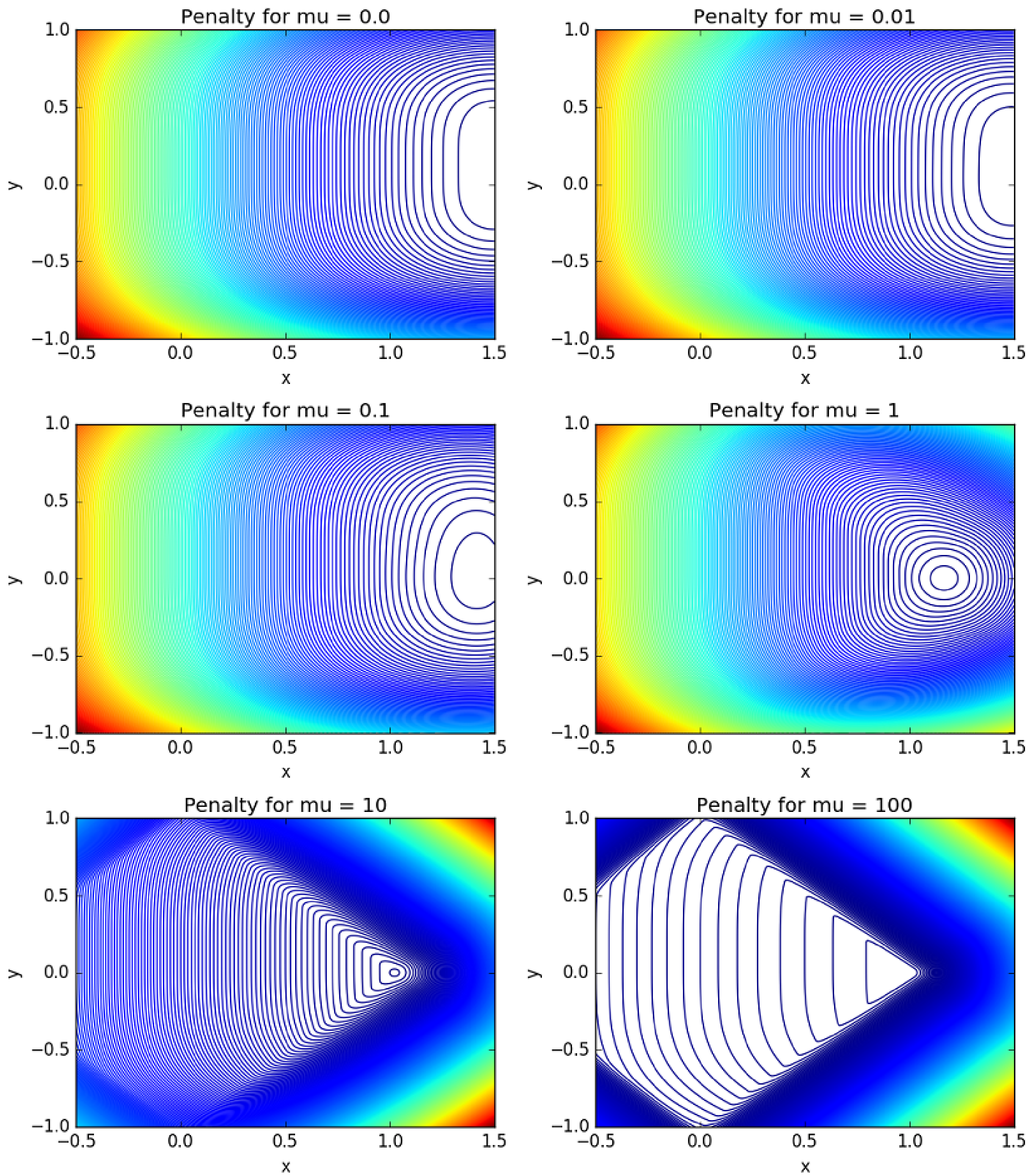

Figure 9: The influence of the penalty parameter on the objective. As μ grows, the feasible set is more and more obvious in the function values.

```
## Armijo parameters:
alpha0 = 1.0; beta = 0.5; c = 1e-1;
## See the Armijo linesearch function in previous examples.

## Starting point
x_SD = [-0.5;0.8];

muVec = [0.1; 1.0 ; 10.0 ; 100.0]
fvals = []; cvals = [];figure();
## SD Iterations
for jmu = 1:length(muVec)
    mu = muVec[jmu]
    fmu = (x)->f(x,mu);
    println("mu = ",mu)
    Fc = F + mu*C;
    subplot(2,2,jmu); contour(X,Y,Fc,1000); #hold on; axis image;
    xlabel("x"); ylabel("y"); title(string("SD for mu = ",mu))
    for k=1:50
        gk = g(x_SD,mu);
        if norm(gk) < 1e-3
            break;
        end
        d_SD = -gk;
        x_prev = copy(x_SD);
        alpha_SD = linesearch(fmu,x_SD,d_SD,gk,0.5,beta,c);
        x_SD=x_SD+alpha_SD*d_SD;
        plot([x_SD[1];x_prev[1]],[x_SD[2];x_prev[2]],"g",linewidth=2.0);
        println(x_SD,",",norm(gk),",",alpha_SD,",",f(x_SD,mu));
        fvals = [fvals;f(x_SD,mu)];
        cvals = [cvals;norm(max(A*x_SD - oVec,0.0))]
    end;
end
figure();subplot(1,2,1);
plot(fvals);title("function values")
subplot(1,2,2);
semilogy(cvals,"*");title("Constraints violation values ||max(Ax-1,0)||")
```

Barrier methods We will now see the "sister" of the penalty method which offers a different penalty approach. This approach is relevant only for inequality constraints, and is used when we wish to require that the iterates will absolutely be inside the feasible domain. The idea is to choose a "barrier" function that goes to ∞ as we move towards the boundaries of the feasible domain. Unlike the penalty approach we do not even define the objective outside the feasible domain, and do not allow our iterates to go there.

There are two main "barrier functions": the log-barrier function and the inverse-barrier

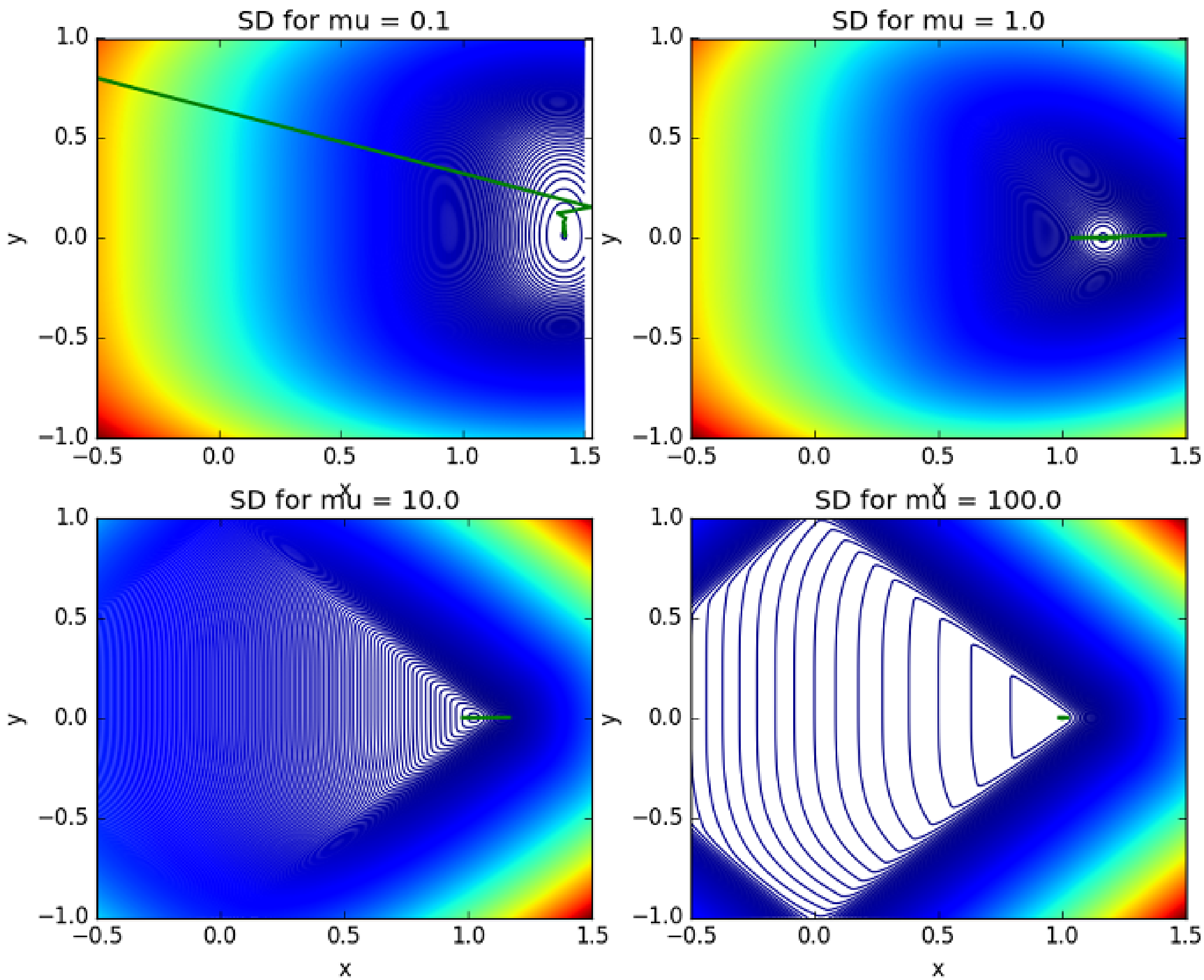

Figure 10: The convergence paths of the SD method for minimizing the objective for various penalty parameters. Each solution is initialized with the solution for the previous larger penalty parameter μ.

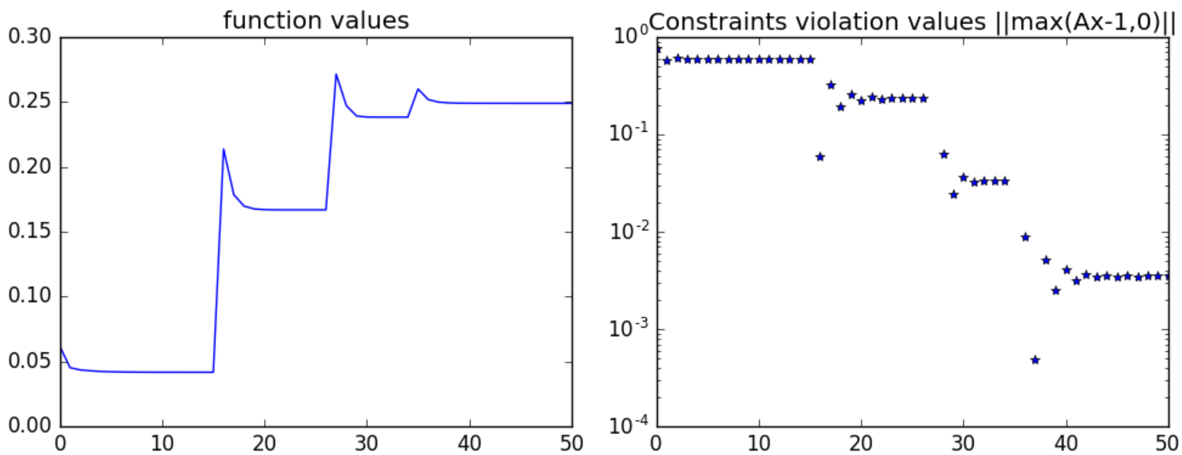

Figure 11: The function value and constraints violation for the SD iterations. At $\mu = 100$ we get about three digits of accuracy for the constraints.

function. Suppose we wish to have a constraint $\phi(x) \leq 0$, then the two following functions can be used:

$$b_{log}(x) = -\log(\phi(x)), \quad b_{inv}(x) = -\frac{1}{\phi(x)}$$

Figure 12 shows an example of the two penalty functions. Using these functions, the minimizer will only be inside the domain and as we apply the iterates, we have to make sure that our next step is always inside the domain. If not, we reduce the steplength parameter until the next iterate is inside the domain (say, in an iterative matter similar to the Armijo process). Since these functions go to ∞, the barrier method is not too sensitive to the choice of penalty parameter, but note that 1) these barrier functions are not 0 inside the domain, and hence influence the solution even if the constraint is not active. 2) These are highly non-quadratic functions with quite large high order derivatives (third derivative for example). The optimization process which works best on quadratic penalties usually struggles with these functions.

3.4 The projected Steepest Descent method

The projected Steepest Descent method is an attempt to avoid all the difficulties involved with the penalty and barrier methods. In this method we assume that the iterates $\mathbf{x}^{(k)}$ are inside or on the boundaries of the feasible domain. In this method we find a Steepest Descent

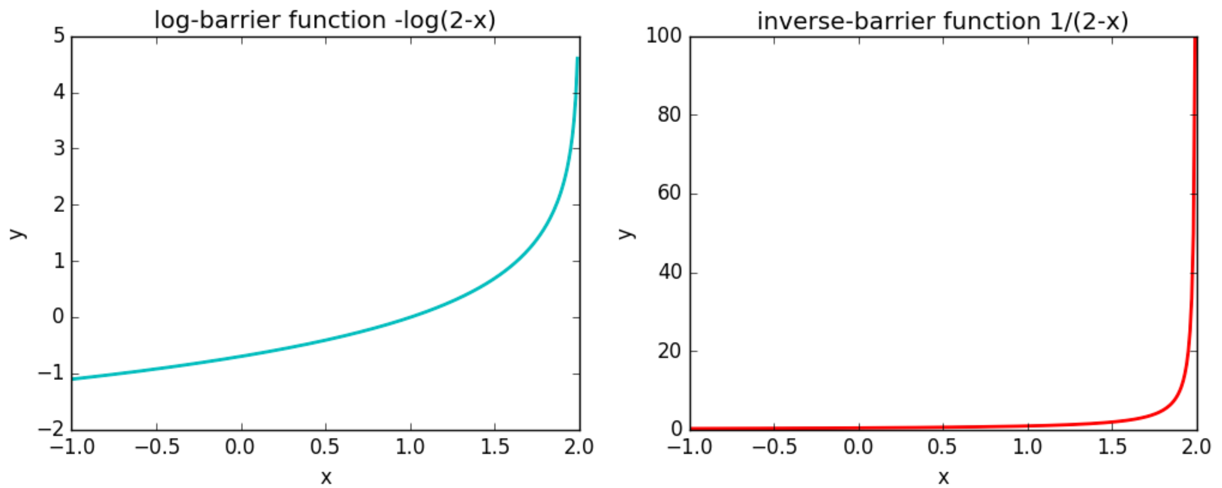

Figure 12: Two barrier functions for the constraint $x - 2 \leq 0$. Both of the functions go to ∞ as $x \to 0$. The log function is bounded from below only in a closed region, so it is not suitable for cases where only one sided barrier is used.

step $\mathbf{y}$, followed by a projection of $\mathbf{y}$ to be inside the feasible domain. That is, we replace the $\mathbf{y}$ with the closest point that fulfils the constraints. This is not always an easy task, therefore this method is most useful when the constraints are simple (linear, for example).

More explicitly, assume the problem

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$$

where Ω is a sub-set of $\mathbb{R}^n$. Given a point $\mathbf{y}$, we will define the projection operator Π_Ω to be the solution of the following problem:

$$\Pi_\Omega(\mathbf{y}) = \arg\min_{\mathbf{x}\in\Omega} \|\mathbf{x} - \mathbf{y}\|,$$

for some vector norm (in most cases the squared ℓ_2 norm is chosen).

The projected SD method is defined by

$$\mathbf{x}^{(k+1)} = \Pi_\Omega \left(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)})\right),$$

where α is obtained by linesearch.

Example 12. *Define the projected steepest descent method for the linearly constrained minimization*

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \quad \textit{subject to} \quad A\mathbf{x} = \mathbf{b},$$

assuming $A \in \mathbb{R}^{m\times n}$ *with* $m < n$ *is full rank).*

All we need to define is the projection operator with respect to some norm, and then set $\mathbf{x}^{(k+1)} = \Pi_\Omega\left(\mathbf{x}^{(k)} - \alpha\nabla f(\mathbf{x}^{(k)})\right)$*. We will choose the squared* ℓ_2 *norm.*

$$\Pi_\Omega(\mathbf{y}) = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2, \quad \textit{subject to} \quad A\mathbf{x} = \mathbf{b}$$

The Lagrangian of the system is given by

$$\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \boldsymbol{\lambda}^\top(A\mathbf{x}-\mathbf{b}).$$

To solve the problem we will solve the system

$$\nabla_{\mathbf{x}}\mathcal{L} = \mathbf{x} - \mathbf{y} + A^\top\boldsymbol{\lambda} = 0 \Rightarrow \mathbf{x}^* = \mathbf{y} - A^\top\boldsymbol{\lambda}.$$

The lagrange multiplier $\boldsymbol{\lambda}^*$ *will be defined by the constraint:*

$$A\mathbf{x}^* = \mathbf{b} \Rightarrow A(\mathbf{y} - A^\top\boldsymbol{\lambda}) = \mathbf{b} \Rightarrow \boldsymbol{\lambda}^* = (AA^\top)^{-1}(A\mathbf{y}-\mathbf{b}).$$

Here we see that we should invert $AA^\top \in \mathbb{R}^{m\times m}$*, which is invertible because we assumed that* A *is full rank. To get the final solution for the projection we set*

$$\mathbf{x}^* = \mathbf{y} - A^\top\boldsymbol{\lambda}^* = \mathbf{y} - A^\top(AA^\top)^{-1}(A\mathbf{y}-\mathbf{b}).$$

Now we set $\mathbf{y} = \mathbf{x}^{(k)} - \alpha\nabla f(\mathbf{x}^{(k)})$*, and assume that* $\mathbf{x}^{(k)}$ *satisfies the constraint (*$A\mathbf{x}^{(k)} = \mathbf{b}$*):*

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - \alpha\nabla f(\mathbf{x}^{(k)}) - A^\top(AA^\top)^{-1}(A(\mathbf{x}^{(k)} - \alpha\nabla f(\mathbf{x}^{(k)})) - \mathbf{b}) \\ &= \mathbf{x}^{(k)} - \alpha(I - A^\top(AA^\top)^{-1}A)\nabla f(\mathbf{x}^{(k)}).\end{aligned}$$

This way we get the projected SD method. α *is chosen by linesearch. The operator* $(I - A^\top(AA^\top)^{-1}A)$ *is called an orthogonal projection operator. Using this method, every step*

$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ *will be in the null space of* A*, that is* $A(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = 0$.

Example 13. *The box-constrained minimization*

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \quad \textit{subject to} \quad \mathbf{a} \leq \mathbf{x} \leq \mathbf{b},$$

where the bound vectors satisfy: $\mathbf{a} < \mathbf{b}$.

This time, the constraints impose a very simple solution to the problem. The lagrangian is given by

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \boldsymbol{\lambda}_1^\top(\mathbf{x} - \mathbf{b}) + \boldsymbol{\lambda}_2^\top(-\mathbf{x} + \mathbf{a}).$$

and its gradient is given by

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{x} - \mathbf{y} + \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2.$$

But here, there are active and inactive constraints that we need to incorporate. If no constraint is active we will get $\mathbf{x}^* = \mathbf{y}$. *The problem is separable, so if* $a_i \leq y_i \leq b_i$ *then we can set* $x_i^* = y_i$ *without breaking the constraint, and hence* $(\lambda_1^*)_i = (\lambda_2^*)_i = 0$*, because the constraints are inactive. If* $y_i < a_i$*, then the lower bound constraint is active and the upper bound is not. We set* $x_i^* = a_i$ *and*

$$x_i^* - y_i - (\boldsymbol{\lambda}_2)_i = 0 \Rightarrow (\boldsymbol{\lambda}_2^*)_i = a_i - y_i > 0.$$

We get a positive Lagrange multiplier, which is what needs to be. If $y_i > b_i$*, then the upper bound constraint is active and the lower bound is not. We set* $x_i^* = b_i$ *and*

$$x_i^* - y_i + (\boldsymbol{\lambda}_1)_i = 0 \Rightarrow (\boldsymbol{\lambda}_1^*)_i = y_i - b_i > 0.$$

Overall, the projected steepest descent step is given by:

$$\mathbf{z} = \mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}), \quad x_i^{(k+1)} = \begin{cases} a_i & z_i < a_i \\ b_i & z_i > b_i \\ z_i & \textit{otherwise} \end{cases}.$$