ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Pills of numerical optimization

Computational imaging
2024-25

Elena Loli Piccolomini

Dipartimento di Informatica - Scienza e Ingegneria (DISI)

Constrained and unconstrained minimization

Short bibliography:

- Nocedal, Wright, *Numerical Optimization*
- Beck, *Proximal Methods in Optimization*

Constrained and unconstrained minimization

The problem:

$$\min_{x\in D} f(x)$$

- Unconstrained minimization : $D = R^n$
- Constrained minimization $D \subset R^n$

Examples:

1. Box constraints: $a_i \leq x_i \leq b_i$

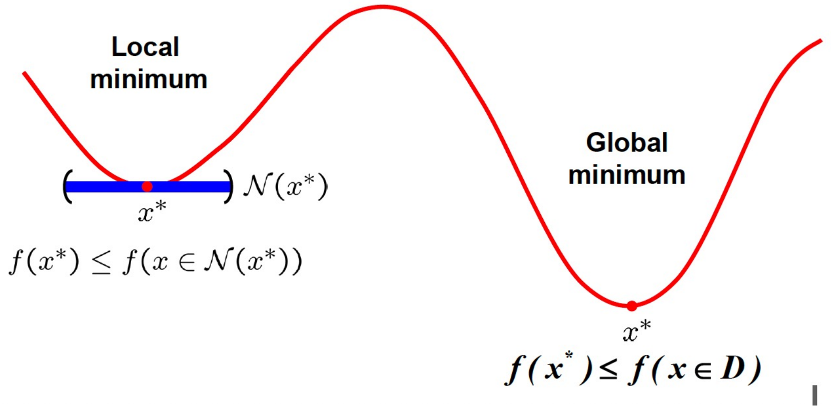
2. positiveness: $x_i > 0, \quad \forall i$

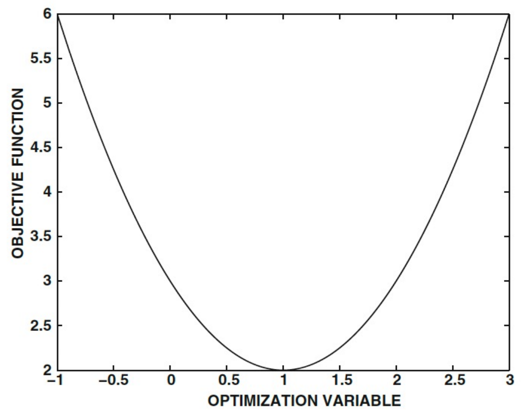
Local and global minima

Consider the problem: $\min_{x \in D} f(x)$

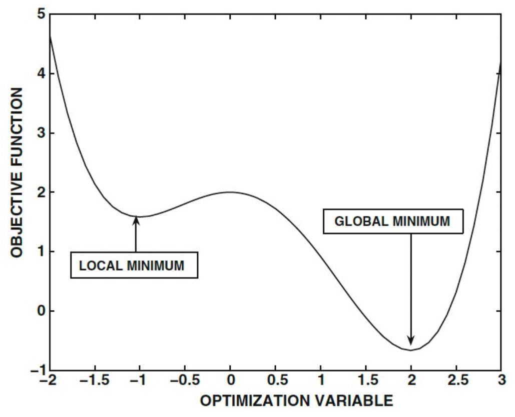
Definitions of local and global minimum

Local and global minima



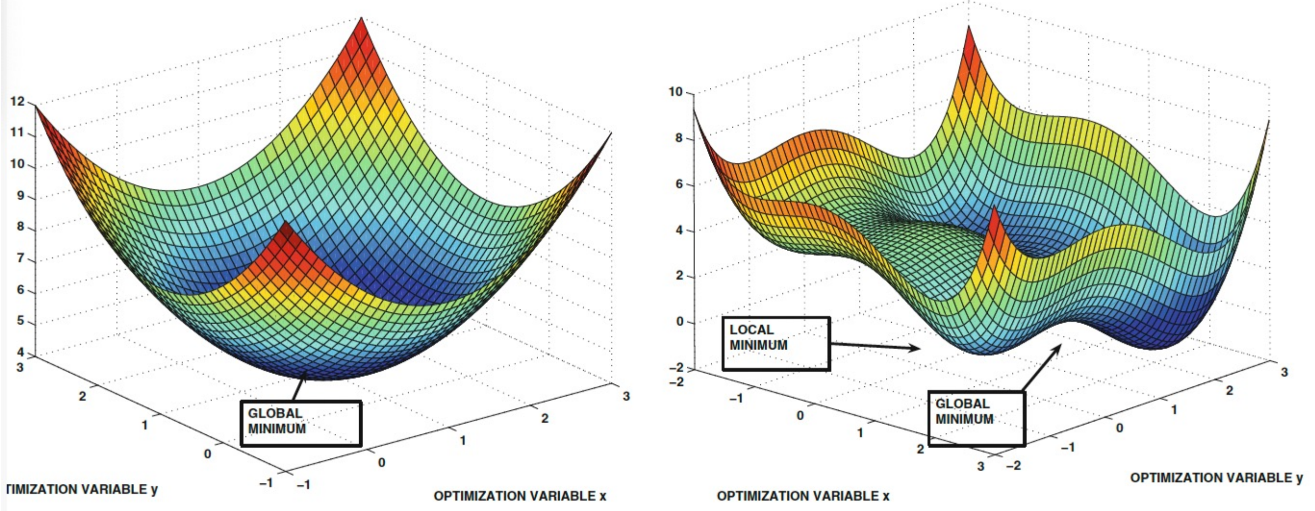


(a) Single global minimum
$f(x) = x^2 - 2x + 3$



(b) Global and local minimum
$F(x) = (x^4/4) - (x^3/3) - x^2 + 2$

Local and global minima

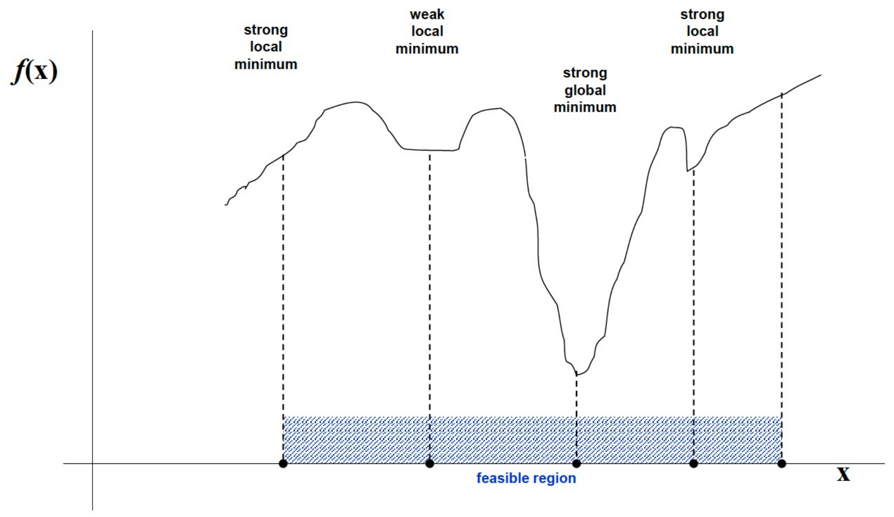


(a) Single global minimum
$g(x,y) = x^2 + y^2 - 2x - 2y + 6$

(b) Global and local minimum
$G(x,y) = ([x^4 + y^4]/4)$
$-([x^3 + y^3]/3) - x^2 - y^2 + 4$

Figure 4.4: Illustrations of local and global optima

Local and global minima



Differentiability

Definition. A function $f : R^n \to R$ is said to be *differentiable at a point* $\mathbf{x_0} \in R^n$ if it exists a linear mapping $J : R^n \to R^n$ so that:

$$lim_{h\to 0}\frac{f(\mathbf{x_0}+\mathbf{h}) - f(\mathbf{x_0}) - J(\mathbf{h})}{\mathbf{h}} = 0$$

This means that f can be approximated by a linear mapping around x_0.

Definition. A function $f : R^n \to R$ is said to be *differentiable in a domain* $D \subset R^n$ if it is differentiable at each point of D.

Proposition. if $D \subset R^n$ is an open set, $f : D \to R^n$ and all the partial derivatives of f in x_0 exist and are continuous, then f is differentiable in x_0.

The opposite is not true.

Unconstrained optimization: optimality conditions

First order conditions.

Proposition **Necessary conditions**. If x^* is a point of local minimum and f is function differentiable at x_0 then

$$\nabla f(x^*) = 0$$

The point x^* is called stationary point.

The condition is NOT sufficient.

Optimality conditions

First order conditions.

Proposition **Necessary conditions**. If x^* is a point of local minimum and f is function differentiable at x_0 then

$$\nabla f(x^*) = 0$$

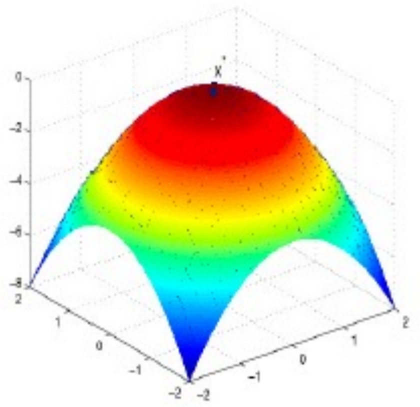
The point x^* is called stationary point.

The condition is NOT sufficient.

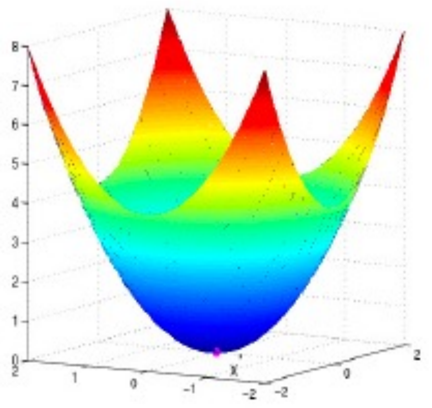
Remark. If x^* is a stationary point for f then it can be:

1. A local minimum point
2. A local maximum point
3. A saddle point

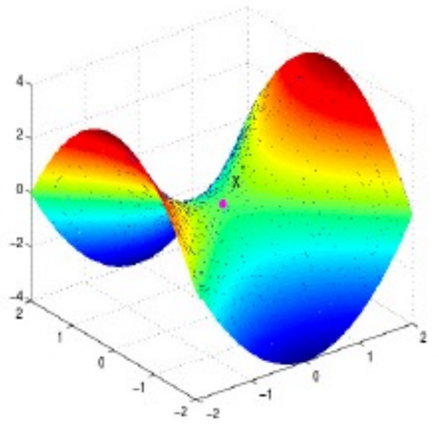
Optimality conditions



Maximum point



Minimum point



Saddle point

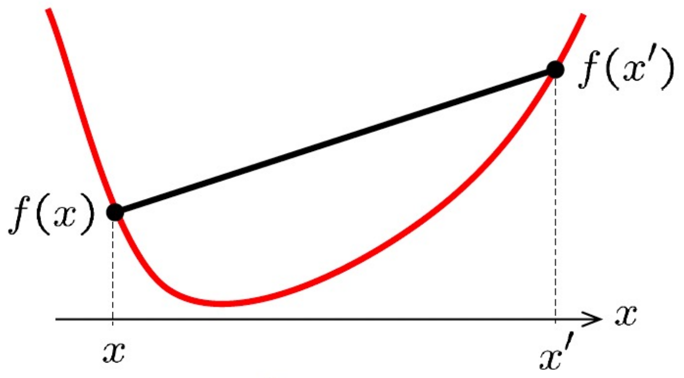
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Convex functions

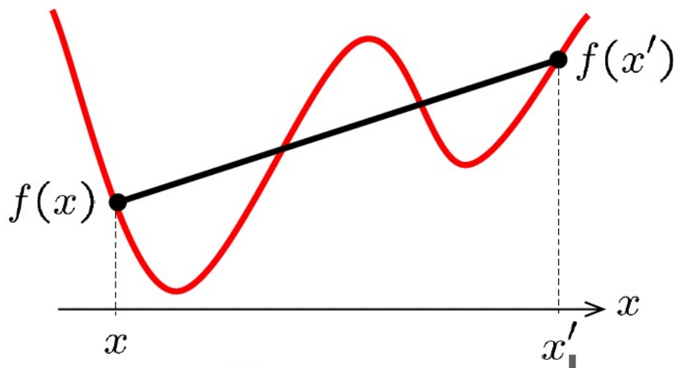
Definition. A function $f : R^n \rightarrow R$ is *convex* if:

$$f(\alpha x + (1 - \alpha y) \leq \alpha f(x) * (1 - \alpha) f(y)$$

$\forall \alpha \in R$ and $\forall x, y \in R^n$.



Convex



Non-convex

Convex functions

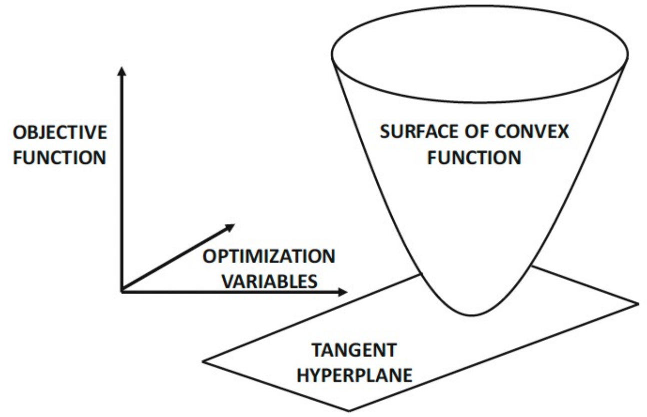


Figure 4.7: A convex function always lies entirely above any tangent to the surface. The example illustrates a 2-dimensional function, where the two horizontal axes are the optimization variables and the vertical axis is the objective function value

Proposition.

- If f is convex, any local minimum of is also a global minimum.
- If f is strictly convex, the f has a unique global minimum.
- If f is convex and differentiable, then any stationary point if f is a global minimum of f.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Convex functions

Examples of convex functions:

- $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$
- $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{y}\|_2^2$, where $A \in R^{n \times m}$, $\mathbf{x} \in R^n$, $\mathbf{y} \in \mathbf{R^m}$

Iterative methods

Given a starting iterate x_0 the iterative method computes a sequence of approximating solutions $x_k = g(x_{k-1}, k = 1, 2, \ldots$, where g is a function or a method to compute k-th iterate as a function of k-1 iterate. The sequence must have the property:

$$lim_{k \to \infty} x_k = x^*$$

where x^* is the solution of the problem.

The implementation of an iterative method requires the definition of stopping criteria.

An important property of the iterative method is the *convergence speed* which is defined

By means of the *order of convergence*.

Iterative methods

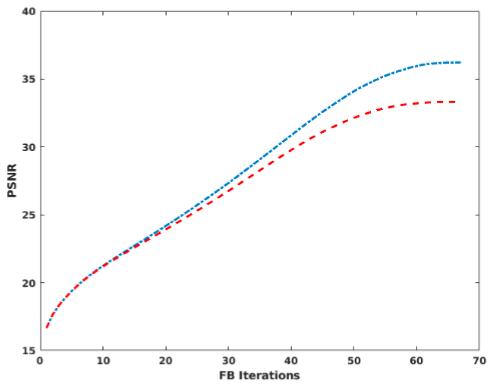
A convergent iterative method is said of order p if:

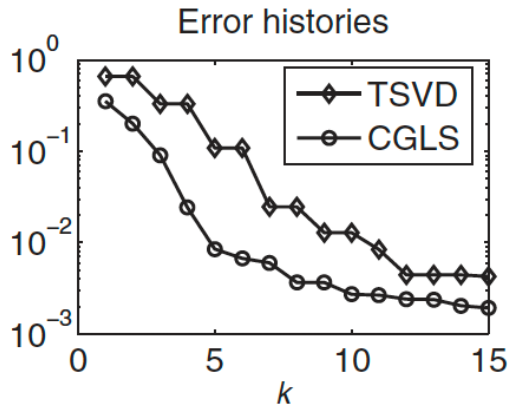
$$\|x_k - x^*\| \leq C\|x_{k-1} - x^*\|^p, \text{ for } k \geq k^*, \quad 1 \leq p \leq 2$$

When k is sufficiently large we can suppose that $\|x_k - x^*\| < 1$.

Hence greater is p faster is the convergence. We also remark that if p=1, then we need C<1 to decrease the error along the iterations.

An heuristic way of analysing the convergence in a test problem is a plot error vs. iterations.





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A general minimization algorithm: gradient descent

Now we consider a general minimization problem

$$min_x f(x)$$

And we suppose f differentiable in its domain.

A general class of minimization algorithms is constutted by *gradient descent methods,* where at **each iteration the function decreases**.

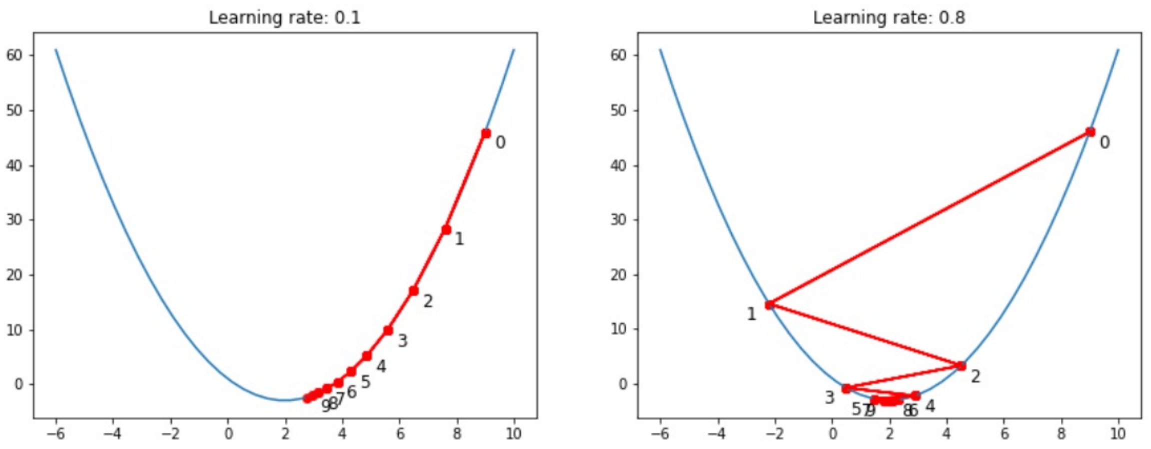
The iteration is expressed as:

$$x_{k+1} = x_k - \alpha_k g_k$$

where:

- $g_k = \nabla f(x_k)$ is the descent direction;
- $\alpha_k \in R^+$ is the step length

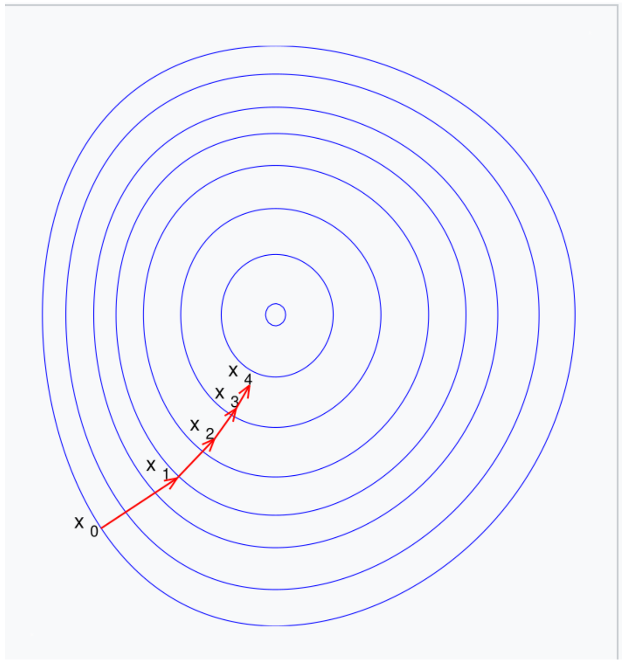
A simple general minimization algorithm: gradient descent



First 10 steps taken by GD for small and big learning rate; Image by author

Case n=1. Iterates of gradient descent method with different values of the step lenghts (learning rate).

A general minimization algorithm: gradient descent

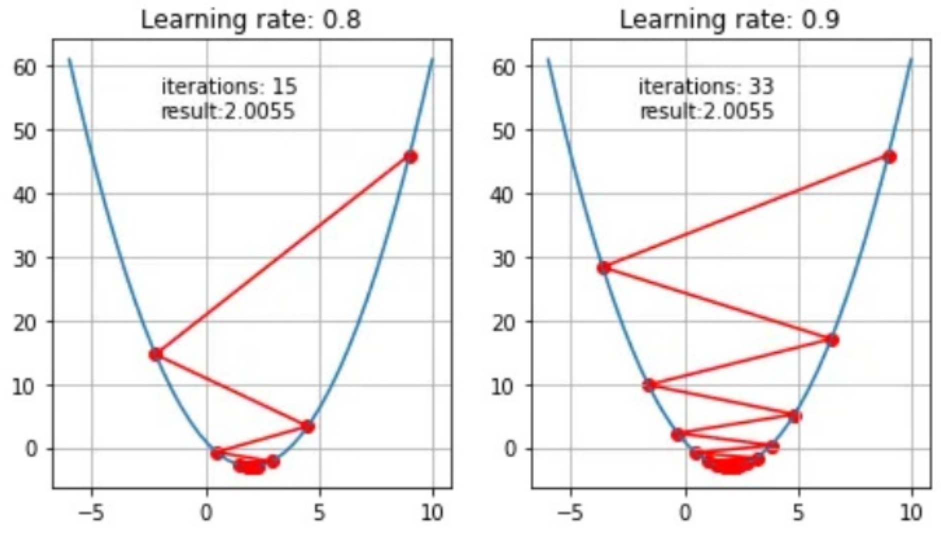


Case n=2. Iterates of gradient descent method with respect to the level curves (corresponding to f(x)=c, c constant).
The gradient in x_k is orthogonal to the tangent to the level curve in x_k.

A general minimization algorithm: gradient descent

Choice of the step length alpha_k.

1. The step length chosen constant (<1). In this case the convergence of the method Is not guaranteed. In generale, a too large step length can cause divergence of the method, whereas a too small step length slows the method convergence (many iterations are required).



A general minimization algorithm: gradient descent

Choice of the step length alpha_k.

2. By means of the backtracking algorithm.

Backtracking algorithm.

1. Initialize $\alpha = 1$
2. Check the Armijo condition:

$$f(x_k - \alpha g_k) \leq f(x_k) - c_1 \alpha_k \|g_k\|_2^2$$

3. if not satisfied set $\alpha = \alpha/\tau$ (usually $\tau = 1/2$ or $\tau = 1/4$)

output: α

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A general minimization algorithm: gradient descent

Theorem. If the step length of the gradient descent method is chosen with a backtracking algorithm, the method converges to a **stationary point(i.e a local minumum or a saddle point).**

The convergence of the gradient descent method is generally very slow, i.e. many iterations are required to compute an accurate solution. There are acceleration techniques, such as the *momentum*, where the computation of the new iterate depends on the last **two** iterations:

$$x_{k+1} = x_k + \gamma(x_k - x_{k-1})$$

The number of iterations is proportional to the computational time.

Each iteration has a computational complexity of the matrix-vector products (in the gradient computation).

Gradient descent: stopping conditions

- The natural stopping condition for the gradient descent method is related to the first order optimality conditions:

$$\|\nabla f(x_k)\|_2 \leq \tau$$ Absolute criterium

$$\|\nabla f(x_k)\|_2 \leq \tau \|\nabla f(x_0)\|_2$$ Relative criterium

- Another stopping condition is related to the distance between two successive iterates:

$$\|x_{k+1} - x_k\|_2 \leq \epsilon$$

- Finally we can fix a number of iterations, but in this case there is no guarantee of having reached the convergence.

$$K = MAXIT$$

Gradient descent: algorithm

Constrained optimization: projected gradient descent

The constrained minimization:

$$min_{a \leq x \leq b} f(x)$$

Where the solution has a **box constraint,** in the sense that it is required to be in the interval [a,b].

In the gradient algorithm, after having computed the iterate x_k , we project it onto the inetrval [a,b]:

$$x_k^+ = \begin{cases} x_k & a \leq x_k \leq b \\ a & x_k < a \\ b & x_k > b \end{cases}$$

Constrained optimization: projected gradient descent

A common example is the case when the solution is required to be non negative is very frequent in imaging, where the image values are physically non negative quantities.

In this case the constraint is simply $x \geq 0$.

In this case after the computation of the iterate x_k, it is projected as:

$$x_k^+ = \begin{cases} x_k & x_k > 0 \\ 0 & \text{otherwise} \end{cases}$$

The convex least squares problem

The linear least squares problem:

$$min_{\mathbf{x}\in R^n} \|A\mathbf{x}-\mathbf{y}\|_2^2$$

$A \in R^{n\times m}$, $\mathbf{x} \in R^n$, $\mathbf{y} \in \mathbf{R^m}$ Suppose m>=n.

The objective function is strictly convex if A has maximum rank n.
Hence the problem has a unique solution if **rank(A)=n,** it has infinitely many solutions if **rank(A)<n.**

Suppose **rank(A)=n**. The unique solution can be computed solving the normal equations:

$$A^T A x = A^T y$$

By means of:
1. Cholesky decomposition (direct method). Used for small and medium size systems

2. Conjugate Gradient Least Squares method (CGLS) that is directly applied to the Least squares function.

The conleast squares problem

Tikhonov regularization:

$$min_{\mathbf{x}\in R^n} \|A\mathbf{x} - \mathbf{y}^{\delta}\|_2^2 + \lambda \|Lx\|_2^2$$

is again a quadratic least squares problem.

We can see it as: $min_{\mathbf{x}\in R^n} \|M\mathbf{x} - \mathbf{y}^{\delta}\|_2^2$

Where $M = \begin{pmatrix} A \\ L \end{pmatrix}$

The normal equations are: $(A^T A + \lambda L^T L)x = A^T y^{\delta}$

The matrix $A^T A + \lambda L^T L$ is symmetric and positive definite (supposed that A and L have maximum rank).

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CGLS algorithm

WE consider here the case of the simple least squares problem where the matrix is A.
In case of Tikhonov regularization it is sufficinet to substitute $A^T A + \lambda L^T L$ in place of A.

The CGLS algorithm is characterized by the fact that it uses the matrix A and A^T
only in matrix-vector operations of the form A*x or A^T*w.
This guarantees that it is not necessary to store the matrix A,
but it can be efficiently computed the matrix vector product, depending on the characteristics of A.

```
Input: $A$, $y$, $x_0$
Compute:
$r_0 = A^T(y - Ax_0)$
$p_0 = r_0$
Repeat for $k = 1, 2, \dots$ until stopping conditions
$q_k = Ap_k$
$\alpha_k = \frac{\|q_k\|_2^2}{\|p_k\|_2^2}$
$x_{k+1} = x_k + \alpha_k p_k$
$r_{k+1} = r_k - \alpha_k A^T q_k$
$\beta_k = \frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2}$
$p_{k+1} = r_k + \beta_k p_k$
```

CGLS algorithm

Stopping conditions.

They are based on a relative estimate of the residual of the system

$$||r_k||_2 < \texttt{tol} \cdot ||r_0||_2.$$

Where **tol** is an input parameter of the calling function.

Usually, a maximum numer of iterations is specified to avoid infinite or too expensive loops.

Even if in exact arithemetic we prove that CGLS converges in at most n iterations to the exact solution,
We van’t say anything in finite arithemtic with rounding errors.

Hence, when the algorithms stops for the maximum number of iterations, we have no guarantees that The computed solution approximates the exact one.

Non differentiable minimization: proximal algorithms

- When the function f is not differentiable it is not possible to apply the gradient descent method
Which requires the computation of the gradient of the function.

- Alternatives algorithms are applied, such as the general class of proximal algorithms.

- Proximal algorithms are a suitable tool to solve large, non-smooth, unconstrained or constrained,
 minimization problems.

- In these methods the computation of the gradient of the non differentiable function
is «substituted» by the so called proximal operator. The comptation of the proximal operator involves itself
A simple convex minimization problem that often has a closed form solution or can be solved very fast
by simple optimization methods.

- Examples are:
  - Chambolle-Pock method
  - Proximal gradient method (Forward-Backward splitting)
  - Iterative Shrinkage-Thresholding method (ISTA/FISTA)

The Total Variation minimization problem

The Total Variation:

$$TV(x) = \sum_{i=1}^{n} \sqrt{(D_h x)_i^2 + (D_v x)_i^2}$$

Is a **nonnegative**, **convex**, **non-differentiable** function in (0,0) (since the ||x||_2 is not differentiable in 0).

To solve the convex minimization problem:

$$min_{x\geq 0} \|Ax - y^\delta\|_2^2 + \lambda \|Dx\|_1$$

We need an algorithm for the minimization of non-convex functions.

Sometimes the TV function is approximated by the *smoothed* TV_β obtaied *by* introducing a small parameter β :

$$TV_\beta(x) = \sum_{i=1}^{n} \sqrt{(D_h x)_i^2 + (D_v x)_i^2 + \beta^2} \qquad \beta > 0$$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The Chambolle-Pock algorithm

- The Chambolle-Pock algorithm does not require the computation of the gradient of The objective function.Instead, it makes use of the proximal (or subdifferential) operators.

Let $f : R^n \to R$ a convex function. Then the proximal operator of f is defined as:

$$prox_{\lambda f}(v) := \arg\min_x (f(x) + \frac{1}{2\lambda} \|x - v\|_2^2)$$

Where $prox_{\lambda f} : R^n \to R^n$ and $\lambda > 0$

- f may be non-smooth
- To evaluate the prox we solve a convex regularization minimization problem
- Often there are analytical (closed-form solutions) to the prox problem.
- The prox is a point that is a compromise between minimizing f and being close to v
- A collection of proximity operators implemented in Matlab and Python: http://proximity-operator.net/

Non differentiable and non convex minimization: TpV

The Total p_Variation TpV

$$TpV(x) = \|Dx\|_p^p = \sum_{i=1}^n (\sqrt{(D_h x)_i^2 + (D_v x)_i^2})^p$$

Is a **non-negative**, **non-convex**, **non–differentiable** function.

This means that the minimization problem:

$$min_{x\geq 0}\|Ax - y^\delta\|_2^2 + \lambda TpV(x)$$

Can have more than one local minima.

The Chambolle-Poack algorithm can be applied to compute a **local minimum** of the objective function.

We do not have any guarantee that the computed minimum is global (or near the global one).