Nonlinear Optimization

Hongjue Zhao

1 The World of Nonlinear Optimization

1.1 General Formulation of the Problem

Let $\boldsymbol{x}$ be an n-dimensional *real vector*

$$\boldsymbol{x} = \left(x^{(1)}, \ldots, x^{(n)}\right)^\top \in \mathbb{R}^n$$

and $f_0(\cdot), \ldots, f_m(\cdot)$ be some *real-valued* functions defined on a set $Q \subset \mathbb{R}^n$. In this way, let's consider the general minimization problem:

$$\begin{aligned} &\min\ f_0(\boldsymbol{x}), \\ &\text{s.t. } f_j(\boldsymbol{x})\ \&\ 0, \quad j = 1, \ldots, m, \\ &\boldsymbol{x} \in Q, \end{aligned} \tag{1.1}$$

where the sign & can be $\leq$, $\geq$ or $=$.

Notations:

- f_0 is called *objective* function.
- The vector function $\boldsymbol{f}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})^\top\right)$ is called the vector of *functional constraints*.
- The set Q is called the *basic feasible set*.
- The set $\mathscr{F} = \{x \in Q \mid f_j(\boldsymbol{x}),\ j = 1, \ldots, m\}$ is called *the entire feasible set* of problem 1.1.

Classification:

1. **Natural Classification**:
   - *Constrained problems*: $\mathscr{F} \subset \mathbb{R}^n$.
   - *Unconstrained problems*: $\mathscr{F} \equiv \mathbb{R}^n$.
   - *Smooth problems*: all $f_j(\cdot)$ are differentiable.
   - *Nonsmooth problems*: there are several nondifferentiable components $f_k(\cdot)$.
   - *Linearly constrained problems*: the functional constraints are affine:

     $$f_j(x) = \langle \boldsymbol{a}_j, \boldsymbol{x} \rangle + b_j$$

     - *Linear optimization Problem*: $f_0(\cdot)$ is also affine.
     - *Quadratic optimization problem*: $f_0(\cdot)$ is Quadratic.
     - *Quadratic constrained quadratic problem*: $f_0(\cdot), \ldots, f_m(\cdot)$ are all quadratic.
2. **Based on the Feasible Set**:

- Problem 1.1 is called *feasible* if $\mathscr{F} \neq \emptyset$.
- Problem 1.1 is called *strictly* feasible if there exists an $\boldsymbol{x} \in Q$ such that $f_j(\boldsymbol{x}) < 0$ for all inequality constraints and $f_j(\boldsymbol{x}) = 0$ for all equality constraints. (*Slater condition.*)

3. **Based on Solution**:

   - A point $\boldsymbol{x}^* \in \mathscr{F}$ is called the optimal *global solution* to problem 1.1 if $f_0(\boldsymbol{x}^*) \leq f_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathscr{F}$ (*global minimum*). $f_0(\boldsymbol{x}^*)$ is called the global *optimal value* of the problem.
   - A point $\boldsymbol{x}^* \in \mathscr{F}$ is called a *local solution* to problem 1.1 if there exists a set $\hat{\mathscr{F}} \subset \mathscr{F}$ such that $\forall \boldsymbol{x} \in \text{int}\hat{\mathscr{F}},\ f_0(\boldsymbol{x}^*) \leq f_0(\boldsymbol{x})$. If $\forall \boldsymbol{x} \in \hat{\mathscr{F}} \setminus \{\boldsymbol{x^*}\},\ f_0(\boldsymbol{x}^*) \leq f_0(\boldsymbol{x})$, then $\boldsymbol{x}^*$ is called *strict* (or *isolated*) local minimum.

Nonlinear Optimization is very import and promising application theory. It covers almost ALL needs of Operation Research and Numerical Analysis. However, in general, optimization problems should be UNSOLVABLE. It is difficult to believe in the existence of a universal tool which is able to solve all problems in the world.

1.2 Performance of Numerical Methods

Usually we focus on the best method for a *class* of problem $\mathscr{P} \ni P$. The *performance* of a method, $\mathscr{M}$ on the whole class $\mathscr{P}$ can be a natural measure of its efficiency. Here we assume that the method $\mathscr{M}$ does not have *complete* information about a particular problem P.

- **Model**: The *model* is the *known* (to a numerical scheme) "part" of problem P and is denoted by Σ. The model consists of:
  - The formulation of the problem.
  - The description of the classes of functional components.
  - etc.
- **Oracle**: The oracle is used to describe the process of collecting this data. A oracle $\mathcal{O}$ is just a unit which answers the successive questions of the methods.

The method $\mathscr{M}$ is trying to solve the problem P by collecting and handling the answers.

For each problem we can develop different types of oracles. But let us fix Σ and $\mathcal{O}$. In this case, it is natural to define the performance of $\mathscr{M}$ on $(\Sigma, \mathcal{O})$ as its performance on the worst P_w from $(\Sigma, \mathcal{O})$. Note that P_w can be only bad for $\mathscr{M}$.

Definition 1.1 (Performance)
The performance of $\mathscr{M}$ on P is the total amount of computational effort *required by the method $\mathscr{M}$ to solve the problem P.* ♣

Notes:

- Solving the problem means finding an *approximate solution* to $\mathscr{P}$ with some accuracy $\epsilon > 0$.
- We use $\mathscr{T}_\epsilon$ to represent a stopping criterion.

Now we have a formal description of the problem class:

$$\mathscr{P} \equiv (\Sigma, \mathcal{O}, \mathscr{T}_\epsilon)$$

In order to solve a problem P from $\mathscr{P}$, we apply it to an *iterative process*.

Algorithm 1: General Iterative Scheme

Input : Starting point $\boldsymbol{x}_0$ and accuracy $\epsilon > 0$.
Output: Solution $\bar{\boldsymbol{x}}$.
Initialization: Set $k = 0$, $\mathscr{I}_{-1} = \emptyset$. Here k is the iteration counter and $\mathscr{I}_k$ is the accumulated *informational set*.
while *True* **do**
  Call oracle $\mathcal{O}$ at point $\boldsymbol{x}_k$.
  Update the informational set: $\mathscr{I}_k = \mathscr{I}_{k-1} \cup (\boldsymbol{x}_k, \mathcal{O}(\boldsymbol{x}_k))$.
  Apply the rules of method $\mathscr{M}$ to $\mathscr{I}_k$ and generate a new point $\boldsymbol{x}_{k+1}$.
  if $\mathscr{I}_\epsilon$ **then**
    Form output $\bar{\boldsymbol{x}}$.
  else
    $k := k + 1$.
  end
end

Now we can specify the meaning of *computational effort* in our definition of performance. In Algorithm 1, we can see two potentially expensive steps:

- In Step 1, where we call the oracle.
- In Step 3, where we form a new test point.

So, we can introduce two measures of complexity of problem P for method $\mathscr{M}$:

Definition 1.2 (Computational Complexity)
Analytical complexity: *The number of calls of the oracle which is necessary to solve the problem P up to the accuracy ϵ.*
Arithmetical complexity: *The total number of arithmetic operations (including the work of oracle and work of method), which is necessary for solving problem P up to accuracy ϵ* ♣

Actually, the second one is more realistic. However, for a particular method $\mathscr{M}$ as applied to problem P, arithmetical complexity can be easily obtained from the analytical complexity and complexity of the oracle.

There is one standard assumption on the oracle which allows us to obtain the majority of results on analytical complexity for optimization schemes. This assumption, called *Local Black Box Concept*, is as follows.

Assumption 1.1 (Local Black Box)

1. *The only information available for the numerical scheme is the* answer of the oracle.
2. *The oracle is* local*: A small variation of the problem far enough from the test point x, which is compatible with the description of the problem class, does not change the answer at x. nonumberplain*

In Assumption 1.1, the first one seems like the artificial wall between the method and the oracle. Although it seems natural to give methods full access to the internal structure of the problem, we can find that when problems have a complicated or implicit structure, this access is almost useless.

Here we conclude problem 1.1 as a *functional model* of optimization problem. According to the degree of smoothness, we can apply different types of oracle:

- Zero-order Oracle: returns the function value $f(\boldsymbol{x})$.
- First-order Oracle: returns the function value $f(\boldsymbol{x})$ and the gradient $\nabla f(\boldsymbol{x})$.
- Second-order Oracle: returns $f(\boldsymbol{x})$, $\nabla f(\boldsymbol{x})$, and the Hessian $\nabla^2 f(\boldsymbol{x})$

1.3 Complexity Bounds for Global Optimization

Let's consider *n-dimensional box problem*:

$$\begin{aligned} &\min_{x \in B_n} \; f(\boldsymbol{x}) \\ &B_n = \{\boldsymbol{x} \in \mathbb{R}^n \mid 0 \le x^{(i)} \le 1, \; i = 1, \ldots, n\} \end{aligned} \tag{1.2}$$

In our terminology, this is a constrained minimization problem without functional constraints. B_n is the basic feasible set, which can be seemed as a n-dimensional box.

Here we use l_∞-norm to measure distances:

$$\|\boldsymbol{x}\|_\infty = \max_{1 \le i \le n} \left| x^{(i)} \right|$$

Then we make the assumption:

Assumption 1.2
the objective function $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ *is* Lipschitz continuous *on* B_n*:*

$$|f(\boldsymbol{x} - \boldsymbol{y})| \le L \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \quad \forall \boldsymbol{x}, \boldsymbol{y} \in B_n,$$

with some constant L *(*Lipschitz constant*).* *nonumberplain*

Let us consider a very simple method for solving problem 1.2, which is called *Uniform Grid Method* $\mathscr{G}(p)$.

Algorithm 2: Uniform Grid Method $\mathscr{G}(p)$

Input : $p \ge 1$
Output: The minimal pair $(\bar{\boldsymbol{x}}, f(\bar{\boldsymbol{x}}))$.
Form p^n points

$$\boldsymbol{x}_\alpha = \left(\frac{2i_1 - 1}{2p}, \frac{2i_2 - 1}{2p}, \ldots, \frac{2i_n - 1}{2p} \right)^\top.$$

where $\alpha \equiv (i_1, \ldots, i_n) \in \{1, \ldots, p\}^n$.
Among all points $\boldsymbol{x}_\alpha$, find the point $\bar{\boldsymbol{x}}$ with the minimal value of the objective function.

Thus, this method forms a uniform grid of the test points in the n-dimensional box, computes the best value of the objective function over the grid, and returns this value as an approximate solution to problem 1.2. In our terminology, this is a zero-order iterative method without any influence from the accumulated information on the sequence of test points. Then let's focus on its efficiency estimate.

Theorem 1.1
Let f^* be a global optimization value of problem 1.2. Then

$$f(\bar{\boldsymbol{x}}) - f^* \leq \frac{L}{2p} \qquad \diamondsuit$$

PROOF (OF THEOREM 1.1) For a multi-index $\alpha = (i_1, \ldots, i_n)$, define

$$X_\alpha = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{x}_\alpha\|_\infty \leq \frac{1}{2p} \right\}$$

Obviously, $\bigcup_{\alpha \in \{1,\ldots,p\}^n} X_\alpha = B_n$.

Let $\boldsymbol{x}^*$ be a global solution of our problem. Then there exists a multi-index α^* such that $\boldsymbol{x}^* \in X_{\alpha^*}$. Note that $\|\boldsymbol{x}^* - \boldsymbol{x}_{\alpha^*}\|_\infty \leq 1/2p$. Therefore,

$$f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}_{\alpha^*}) - f(\boldsymbol{x}^*) \leq \frac{L}{2p} \qquad \blacksquare$$

Let us conclude with the definition of our problem class. We fix our goal as follows:

$$\text{Find } \bar{\boldsymbol{x}} \in B_n : \quad f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^*) \leq \epsilon \tag{1.3}$$

Corollary 1.1
According to 1.2 and 1.1, the analytical complexity of problem class 1.2 for method $\mathscr{G}$ is at most

$$\mathscr{A}(\mathscr{G}) = \left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1 \right)^n . \qquad \spadesuit$$

PROOF (OF COROLLARY 1.1) Take $p = \left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1$. Then $p \geq \frac{L}{2\epsilon}$, and, in view of Theorem 1.1, we have $f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^*) \leq \frac{L}{2p} \leq \epsilon$. Note that we need to call the oracle at p^n points. $\blacksquare$

Thus, $\mathscr{A}(\mathscr{G})$ justifies an *upper* complexity bound for our problem class.

But we still get some questions:

1. It may happen that our proof is too rough and the real performance of method $\mathscr{G}(p)$ is much better.
2. We still cannot be sure that $\mathscr{G}(p)$ is reasonable method for solving problem 1.2. There could exist other schemes with much higher performance.

In order to answer these questions, we need to derive the *lower* complexity bounds for problem class 1.2. The main features of such bounds are as follows.

- They are based on the *Black Box Concept* in Assumption 1.1.
- These bounds are valid for all reasonable iterative schemes. Thus, they provide us with a lower estimate for the analytical complexity of the problem class.

- Very often such bounds employ the idea of a *resisting oracle*.

Resisting Oracle

A *resisting oracle* tries to create the *worst possible* problem for each particular method.

- It starts from an "empty" function and it tries to answer each call of the method in the worst way.
- The answers must be *compatible* with the *previous answers* and with *description of the problem class*.
- After termination of the method, it is possible to *reconstruct* a problem which perfectly fits the final informational set accumulated by the algorithm.
- If we run the method on this newborn problem, it will reproduce the same sequence of test points since it will have the same sequence of answers from the oracle.

EXAMPLE 1.1 Let's consider gradient descent on a Lipschitz-smooth, convex function f (a convex function whose gradient is Lipschitz-continuous). Gradient descent generates a sequence of points $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ which satisfies

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \gamma\nabla f(\boldsymbol{x}_k).$$

Now we are interested in the worst case analysis. We would like to find the "worst" sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}\}$ we could probably get given by gradient descent. But under the assumption that f is convex and Lipschitz-smooth, the resisting oracle here would give you "bad" values of $\nabla f(\boldsymbol{x}_k)$, which lead to slow convergence but are still "possible" in the sense that they are values for which we can construct a convex Lipschitz-smooth function whose gradient evaluated at $\boldsymbol{x}_k$ gives the right values.

Attention: There is no actual function f here. There is only the oracle giving us values when we ask it for $\nabla f(\boldsymbol{x}_k)$. The oracle is constructed in such a way that the values it returns could plausibly be the gradient values of some convex Lipschitz smooth function, but there is no fixed function that we start with. $*$

Theorem 1.2
For $\epsilon < \frac{1}{2}L$, the analytical complexity of problem class $\mathscr{P}_\infty$ is at least $\left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$ calls of oracle. $\diamondsuit$

PROOF (OF THEOREM 1.2) Let $p = \left\lfloor \frac{L}{2\epsilon} \right\rfloor$ (≥ 1). Assume that there exists a method which needs $N < p^n$ calls of oracle to solve any problem. Let us apply this method to the following resisting oracle:

$$\text{Return } f(\boldsymbol{x}) = 0 \text{ at any test point } \boldsymbol{x}.$$

Therefore this method can find only $\bar{\boldsymbol{x}} \in B_n$ with $f(\bar{\boldsymbol{x}}) = 0$.

However, since $N < p^n$, there exists a multi index $\hat{\alpha}$ such that there were no test points in the box $X_{\hat{\alpha}}$. Define $\boldsymbol{x}_* = \boldsymbol{x}_{\hat{\alpha}}$, and consider the function

$$\bar{f}(\boldsymbol{x}) = \min\{0, L\|\boldsymbol{x} - \boldsymbol{x}_*\|_\infty - \epsilon\}.$$

Clearly, this function is l_∞-Lipschitz continuous with constant L, and its global optimal value is $-\epsilon$. Moreover, $\forall \boldsymbol{x} \notin X_{\hat{\alpha}},\ \bar{f}(\boldsymbol{x}) \neq 0$. Thus, $\bar{f}(\cdot)$ is equal to zero *at all test points* of our method (*since we assume that there is no test points in $X_{\hat{\alpha}}$.*)

Since the accuracy of the output of our method is ϵ, we come to the following conclusion: **If the number of calls of the oracle is less than p^n, then the accuracy of the result cannot be better than ϵ.**

Thus, the desired statement is proved. ∎

Theorem 1.3
Let us compare its efficiency estimate with the lower bound:

$$\mathscr{G} : \left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1 \right)^n \Leftrightarrow \text{Lower bound: } \left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$$

If $\epsilon \leq O\left(\frac{L}{n}\right)$, then the lower and upper bounds coincide up to an absolute constant multiplicative factor. This means that, for such level of accuracy, $\mathscr{G}(\cdot)$ is optimal *for the problem class $\mathscr{P}_\infty$.* ♢

Proof (of Theorem 1.3) Since $\epsilon \leq O\left(\frac{L}{n}\right)$, there exists $M > 0$ that satisfies

$$\epsilon \leq M \left| \frac{L}{n} \right| = M \frac{L}{n} \Leftrightarrow \frac{L}{2\epsilon} \geq \frac{n}{2M}.$$

So we can get

$$1 \leq \frac{\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1 \right)^n}{\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor \right)^n} = 1 + \sum_{k=1}^{n} \frac{c_k}{\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor \right)^k} \leq 1 + \sum_{k=1}^{n} \frac{C_n^k}{\left(\left\lfloor \frac{n}{2M} \right\rfloor \right)^k}.$$

As $n \to \infty$, we can also get

$$\lim_{n \to \infty} \left[1 + \sum_{k=1}^{n} \frac{c_k}{\left(\left\lfloor \frac{n}{2M} \right\rfloor \right)^k} \right] = 1.$$

So we can get the result based on *Sandwich theorem*:

$$\lim_{n \to \infty} \frac{\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1 \right)^n}{\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor \right)^n} = 1.$$

Therefore, the statement is proofed. ∎

Theorem 1.2 supports our initial claim that the general optimization problems are *unsolvable*. Let us look the following example.

Example 1.2 Consider the problem class $\mathscr{P}_\infty$ defined by the following parameters: $L = 2$, $n = 10$, $\epsilon = 0.01$. The lower complexity bound for this class is $\left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$ calls of oracle. Let us compute this value for our example.

- **Lower bound**: 10^{20} calls of the oracle
- **Oracle Complexity**: at least n arithmetic operations (a.o.)
- **Total Complexity**: 10^{21} a.o.
- **Processor Performance**: 10^6 a.o. per second
- **Total Time**: 10^{15}s
- **One year**: less than 3.2×10^7 s

- **We need**: $31,250,000$ years. $*$

This estimate is so disappointing that we cannot maintain any hope that such problems may become solvable in the future. The lower complexity bounds for problems with smooth functions, or for high-order methods, are not much better than the bound of Theorem 1.2.

At the end of this section, let us compare our observations with other fields. The uniform grid approach is a standard tool in many domains. Let's consider the numerical integral of a univariate function

$$\mathscr{J} = \int_0^1 f(x)\mathrm{d}x \quad \Rightarrow \quad S_N = \frac{1}{N}\sum_{i=1}^{m} f(x_i).$$

If $f(\cdot)$ is Lipschitz continuous, then the value is good approximation to $\mathscr{J}$:

$$N = L/\epsilon \quad \Rightarrow \quad |\mathscr{J} - S_N| \leq \epsilon.$$

This is a standard way for approximating integrals. The reason why it works here is related to *dimension* of the problem. For integration, the standard dimensions are very small (up to three). However, in optimization, sometimes we need to solve problems with several million variables.

1.4 Identity Cards of the Fields

We try to find a reasonable target in the theoretical analysis of optimization schemes. It seems that everything is clear with general Global Optimization. However:

- Maybe the goals of this field are too ambitious?
- In some practical problems could we be satisfied by much less "optimal" solutions?
- Are there some interesting problem classes which are not as dangerous as the class of general continuous functions?

In fact, each of these questions can be answered in different ways. In the field of nonlinear optimization, they differ one from another in the following aspects:

- Goals of the methods.
- Classes of functional components.
- Description of an oracle.

These aspects naturally define the list of desired properties of the optimization methods, just as follows.

General Global Optimization

- **Goals**: Find a global minimum.
- **Functional Class**: Continuous functions.
- **Oracle**: $0-1-2$ order Black Box
- **Desired Properties**: Convergence to a global minimum.
- **Features**: From theoretical point of view, this game is too short.
- **Problem Sizes**: Sometimes, we can solve problems with many variables. No guarantee of success even for small problems.

General Nonlinear Optimization

- **Goals**: Find a local minimum.
- **Functional Class**: Differentiable functions.
- **Oracle**: 1-st and 2-nd-order Black Box.
- **Desired Properties**: Fast convergence to a local minimum.
- **Features**: Variability of approaches. Most widespread software. The goals are not always acceptable and reachable.
- **Problem Sizes**: Up to several thousand variables.

Black Box Convex Optimization

- **Goals**: Find a global minimum.
- **Functional Class**: Convex sets and functions.
- **Oracle**: 1-st and 2-nd-order Black Box.
- **Desired Properties**: Convergence to a global minimum. The convergence rate may depend on dimension.
- **Features**: Very interesting and rich on complexity theory. Efficient practical methods. The problem class is sometimes restrictive.
- **Problem Sizes**: Several thousand variables for the 2-nd order methods, and several million for the 1-st order schemes.

Structural Optimization

- **Goals**: Find a global minimum.
- **Functional Class**: Simple convex sets and functions with explicit minmax structure.
- **Oracle**: 2-nd-order Black Box for special barrier functions and modified 1-st-order Black Box.
- **Desired Properties**: Fast convergence to a global minimum. The convergence rate may depend on the structure of the problem.
- **Features**: Very new and perspective theory rejecting the Black Box concept. The problem class is practically the same as in Convex optimization.
- **Problem Sizes**: Sometimes up to several million variables

2 Local Methods in Unconstrained Minimization

In this section, we consider several methods for solving the following unconstrained minimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}), \tag{2.1}$$

where $f(\cdot)$ is a smooth function.

2.1 Relaxation and Approximation

The simplest goal in general Nonlinear Optimization consists in *finding a local minimum of a differentiable function*. The majority of methods in general Nonlinear Optimization are based on the idea of *relaxation*.

Definition 2.1 (Relaxation)
A sequence of real numbers $\{a_k\}_{k=0}^{\infty}$ *is called a* relaxation sequence *if*

$$a_{k+1} \leq a_k \quad \forall k > 0.$$ ♣

In order to deal with problem 2.1, most of methods generate a relaxation sequence of function values $\{f(\boldsymbol{x}_k)\}_{k=0}^{\infty}$:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k), \quad k = 0, 1, \ldots .$$

This rule has the following important advantages:

1. If $f(\cdot)$ is bounded below on $\mathbb{R}^n$, then the sequence $\{f(\boldsymbol{x}_k)\}_{k=0}^{\infty}$ converges.
2. In any case, we improve the initial value of the objective function.

In the meanwhile, there is another concept which is essential to implement the idea of relaxation. That is *approximation*.

Definition 2.2 (Approximation)
To approximate means to replace an initial complex object by a simper one which is close to the original in terms of its properties. ♣

In Nonlinear Optimization, we usually apply *local approximation* based on derivatives of nonlinear functions. These are 1-st- and 2-nd-order approximations (or, the linear and quadratic approximations).

Definition 2.3 (Linear Approximation)
Let the function $f(\cdot)$ *be differentiable at* $\bar{\boldsymbol{x}} \in \mathbb{R}^n$. *Then, for any* $\boldsymbol{y} \in \mathbb{R}^n$ *we have*

$$f(\boldsymbol{y}) = f(\bar{\boldsymbol{x}}) + \langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{y} - \bar{\boldsymbol{x}} \rangle + o(\|\boldsymbol{y} - \bar{\boldsymbol{x}}\|),$$

where $\nabla f(\bar{\boldsymbol{x}})$ *is called the* gradient *of the function* f *at* $\bar{\boldsymbol{x}}$. $o(\cdot) : [0, \infty) \to \mathbb{R}$ *is a function of* $r \geq 0$ *satisfying the conditions*

$$\lim_{r \to 0} \frac{1}{r} o(r) = 0, \quad o(0) = 0.$$

♣

Here we use the notation $\|\cdot\|$ for the standard *Euclidean* norm in $\mathbb{R}^n$:

$$\|\boldsymbol{x}\| = \left[\sum_{i=1}^{n} \left(x^{(i)} \right)^2 \right]^{1/2} = (\boldsymbol{x}^\top \boldsymbol{x})^{1/2} = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{1/2},$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product in corresponding coordinate space. Note that

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \ \boldsymbol{y} \in \mathbb{R}^m, \boldsymbol{A} \in \mathbb{R}^{m \times n} \quad \Rightarrow \quad \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y} \rangle \equiv \left\langle \boldsymbol{x}, \boldsymbol{A}^\top \boldsymbol{y} \right\rangle,$$

Gradient: Consider the points: $\boldsymbol{y}_i = \bar{\boldsymbol{x}} + \epsilon \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the i-th coordinate vector in $\mathbb{R}^n$, and taking $\epsilon \to 0$, we can get the following representation of the gradient:

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\bar{\boldsymbol{x}})}{\partial x^{(1)}}, \ldots, \frac{\partial f(\bar{\boldsymbol{x}})}{\partial x^{(n)}} \right)^\top \tag{2.2}$$

Here we mention two important properties of the gradient. Denote by $\mathscr{L}_f(\alpha)$ the *(sub)level set* of $f(\cdot)$:

$$\mathscr{L}_f(\alpha) = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq \alpha\}.$$

Consider the set of directions that are *tangent* to $\mathscr{L}_f(f(\bar{\boldsymbol{x}}))$ at $\bar{\boldsymbol{x}}$:

$$S_f(\bar{\boldsymbol{x}}) = \left\{ \boldsymbol{s} \in \mathbb{R}^n \mid \boldsymbol{s} = \lim_{k \to \infty} \frac{\boldsymbol{y}_k - \bar{\boldsymbol{x}}}{\|\boldsymbol{y}_k - \bar{\boldsymbol{x}}\|}, \text{ for some } \{\boldsymbol{y}_k\} \to \bar{\boldsymbol{x}} \text{ with } f(\boldsymbol{y}_k) = f(\bar{\boldsymbol{x}}) \ \forall k \right\}.$$

Lemmas 2.1
If $\boldsymbol{s} \in S_f(\bar{\boldsymbol{x}})$, *then* $\langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{s} \rangle = 0$. ♡

Proof (of Lemma 2.1) Since $f(\boldsymbol{y}_k) = f(\bar{\boldsymbol{x}})$, we can get

$$f(\boldsymbol{y}_k) = f(\bar{\boldsymbol{x}}) + \langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{y}_k - \bar{\boldsymbol{x}} \rangle + o(\|\boldsymbol{y}_k - \bar{\boldsymbol{x}}\|) = f(\bar{\boldsymbol{x}}).$$

Therefore $\langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{y}_k - \bar{\boldsymbol{x}} \rangle + o(\|\boldsymbol{y}_k - \bar{\boldsymbol{x}}\|) = 0$. So dividing the equation by $\|\boldsymbol{y}_k - \bar{\boldsymbol{x}}\|$ and taking the limit as $\boldsymbol{y}_k \to \bar{\boldsymbol{x}}$, we can obtain the result. ∎

The Fastest Local Decrease

Let $\boldsymbol{s}$ be a direction in $\mathbb{R}^n$, $\|\boldsymbol{s}\| = 1$. Consider the local decrease of the function $f(\cdot)$ along the direction $\boldsymbol{s}$:

$$\begin{aligned}\Delta(\boldsymbol{s}) &= \lim_{\alpha\to 0}\frac{1}{\alpha}[f(\bar{\boldsymbol{x}}+\alpha\boldsymbol{s}) - f(\bar{\boldsymbol{x}})] \\ &= \lim_{\alpha\to 0}\frac{1}{\alpha}[\alpha\langle\nabla f(\bar{\boldsymbol{x}}), \boldsymbol{s}\rangle + o(\alpha)] \\ &= \langle\nabla f(\bar{\boldsymbol{x}}), \boldsymbol{s}\rangle\end{aligned}$$

Based on Cauchy-Schwarz inequality $-\|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\| \le \langle\boldsymbol{x},\boldsymbol{y}\rangle \le \|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\|$, we can obtain that $\Delta(\boldsymbol{s}) = \langle\nabla f(\bar{\boldsymbol{x}}), \boldsymbol{s}\rangle \ge -\|\nabla f(\bar{\boldsymbol{x}})\|$. Let us take $\bar{\boldsymbol{s}} = -\nabla f(\bar{\boldsymbol{x}})/\|\nabla f(\bar{\boldsymbol{x}})\|$, then

$$\Delta(\boldsymbol{s}) = -\langle\nabla f(\bar{\boldsymbol{x}}), \nabla f(\bar{\boldsymbol{x}})\rangle / \|\nabla f(\bar{\boldsymbol{x}})\| = -\|\nabla f(\bar{\boldsymbol{x}})\|.$$

Thus, the direction of $-\nabla f(\bar{\boldsymbol{x}})$ (*the anti-gradient*) is the direction of the *fastest local decrease* of the function $f(\cdot)$ at point $\bar{\boldsymbol{x}}$.

Theorem 2.1 (First-Order Optimality Condition)
Let $\boldsymbol{x}^*$ be a local minimum of a differentiable function $f(\cdot)$. Then

$$\nabla f(\boldsymbol{x}^*) = 0. \tag{2.3}$$

♢

Proof (of Theorem 2.1) Since $\boldsymbol{x}^*$ is a local minimum of $f(\cdot)$,

$$\exists r > 0,\ \forall \boldsymbol{y} \in \mathbb{R}^n\ \|\boldsymbol{y} - \boldsymbol{x}\| < r, \quad \text{s.t. } f(\boldsymbol{y}) \ge f(\boldsymbol{x}^*).$$

Since $f(\cdot)$ is *differentiable*, we can infer that

$$f(\boldsymbol{y}) = f(\boldsymbol{x}^*) + \langle\nabla f(\boldsymbol{x}^*), \boldsymbol{y} - \boldsymbol{x}^*\rangle + o(\|\boldsymbol{y} - \boldsymbol{x}^*\|) \ge f(\boldsymbol{x}^*).$$

Thus $\forall \boldsymbol{s} \in \mathbb{R}^n$, we have $\langle\nabla f(\boldsymbol{x}^*), \boldsymbol{s}\rangle \ge 0$. By taking $\boldsymbol{s} = -\nabla f(\boldsymbol{x}^*)$, we get $-\|\nabla f(\boldsymbol{x}^*\|)^2 \ge 0$. Hence, $\nabla f(\boldsymbol{x}) = 0$. ■

Definition 2.4 (Quadratic Approximation)
Let $f(\cdot)$ be twice differentiable at $\bar{\boldsymbol{x}}$. Then

$$f(\boldsymbol{y}) = f(\bar{\boldsymbol{x}}) + \langle\nabla f(\bar{\boldsymbol{x}}), \boldsymbol{y} - \bar{\boldsymbol{x}}\rangle + \frac{1}{2}\left\langle\nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{x}}), \boldsymbol{y} - \bar{\boldsymbol{x}}\right\rangle + o(\|\boldsymbol{y} - \bar{\boldsymbol{x}}\|^2), \tag{2.4}$$

where $\nabla^2 f(\bar{\boldsymbol{x}})$ is the Hessian *matrix of function f at $\bar{\boldsymbol{x}}$ and $\boldsymbol{o}(\cdot) : [0,\infty)\mathbb{R}^n$ is a continuous vector function satisfying the condition*

$$\lim_{r\to 0}\frac{1}{r}\|\boldsymbol{o}(r)\| = 0.$$

♣

Hessian Matrix

For Hessian matrix $\nabla^2 f(\bar{\boldsymbol{x}})$:

1. $\nabla^2 f(\bar{\boldsymbol{x}})^{(i,j)} = \frac{\partial^2 f(\bar{\boldsymbol{x}})}{\partial x^{(i)} \partial x^{(j)}}$.
2. $\nabla^2 f(\bar{\boldsymbol{x}})$ is a *symmetric* matrix: $\nabla^2 f(\bar{\boldsymbol{x}}) = \left[\nabla^2 f(\bar{\boldsymbol{x}})\right]^\top$.
3. The Hessian can be regarded as a *derivative* of the vector $\nabla f(\cdot)$:

$$\nabla f(\boldsymbol{y}) = \nabla f(\bar{\boldsymbol{x}}) + \nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{x}}) + \boldsymbol{o}(\|\boldsymbol{y} - \bar{\boldsymbol{x}}\|) \in \mathbb{R}^n$$

Based on the quadratic approximation, we can write down the *second-order optimality condition*.

Theorem 2.2 (Second-Order Optimality Condition)
Let $\boldsymbol{x}^*$ be a local minimum of a twice differentiable function $f(\cdot)$. Then

$$\nabla f(\boldsymbol{x}^*) = 0, \quad \nabla^2 f(\boldsymbol{x}^*) \succeq 0. \tag{2.5}$$

♢

Proof (of Theorem 2.2) Since $\boldsymbol{x}^*$ is a local minimum of the function $f(\cdot)$,

$$\exists r > 0, \ \forall \boldsymbol{y} \in \mathbb{R}^n \ \|\boldsymbol{y} - \boldsymbol{x}\| < r, \quad \text{s.t. } f(\boldsymbol{y}) \geq f(\boldsymbol{x}^*).$$

In view of Theorem 2.1, $\nabla f(\boldsymbol{x}^*) = 0$. Therefore, for any such $\boldsymbol{y}$,

$$f(\boldsymbol{y}) = f(\boldsymbol{x}^*) + \left\langle \nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*), \boldsymbol{y} - \boldsymbol{x}^* \right\rangle + o(\|\boldsymbol{y} - \boldsymbol{x}^*\|^2) \geq f(\boldsymbol{x}^*).$$

Thus, $\left\langle \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{s}, \boldsymbol{s} \right\rangle \geq 0$, for all $\boldsymbol{s}$, $\|\boldsymbol{s}\| = 1$. ∎

Again, Theorem 2.2 is a *necessary* (2-nd-order) characteristic of a local minimum. The *sufficient* condition is as follows.

Theorem 2.3
Let a function $f(\cdot)$ be twice differentiable on $\mathbb{R}^n$ and let $\boldsymbol{x}^* \in \mathbb{R}^n$ satisfy the following conditions:

$$\nabla f(\boldsymbol{x}^*) = 0, \quad \nabla^2 f(\boldsymbol{x}^*) \succ 0. \tag{2.6}$$

Then $\boldsymbol{x}^*$ is a strict local minimum of $f(\cdot)$ ♢

Proof (of Theorem 2.3) In a small neighborhood of a point $\boldsymbol{x}^*$ the function $f(\cdot)$ can be represented as

$$f(\boldsymbol{y}) = f(\boldsymbol{x}^*) + \frac{1}{2}\left\langle \nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{y} - \boldsymbol{x}^*), \boldsymbol{y} - \boldsymbol{x}^* \right\rangle + o(\|\boldsymbol{y} - \boldsymbol{x}^*\|^2)$$

Since $\lim_{r \to 0} \frac{o(r^2)}{r^2} = 0$, there exists a value $\bar{r} > 0$ such that for all $r \in [0, \bar{r}]$ we have

$$\left|o(r^2)\right| \leq \frac{r^2}{4} \lambda_{\min}(\nabla^2 f(\boldsymbol{x}^*)).$$

In the view of our assumption, this eigenvalue is *positive*. Therefore, for any $\boldsymbol{y} \in \mathbb{R}^n$, $0 \leq \|\boldsymbol{y} - \boldsymbol{x}^*\| \leq \bar{r}$, we have

$$\begin{aligned} f(\boldsymbol{y}) &\geq f(\boldsymbol{x}^*) + \frac{1}{2}\lambda_{\min}(\nabla^2 f(\boldsymbol{x}^*))\|\boldsymbol{y} - \boldsymbol{x}^*\|^2 + o(\|\boldsymbol{y} - \boldsymbol{x}^*\|^2) \\ &\geq f(\boldsymbol{x}^*) + \frac{1}{4}\lambda_{\min}(\nabla^2 f(\boldsymbol{x}^*))\|\boldsymbol{y} - \boldsymbol{x}^*\|^2 > f(\boldsymbol{x}^*) \end{aligned}$$

■

2.2 Classes of Differentiable Functions

Definition 2.5
Let Q be a subset of R^n. We denote by $C_L^{k,p}(Q)$ the class of functions with the following properties:

- *any $f \in C_L^{k,p}(Q)$ is k times continuously differentiable on Q*
- *its pth derivative is* Lipschitz *continuous on Q with constant L:*

$$\|\nabla^p f(\boldsymbol{x}) - \nabla^p f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x},\ \boldsymbol{y} \in Q$$

♣

Clearly we always have $p \leq k$. If $q > k$, then $C_L^{q,p}(Q) \subset C_L^{k,p}(Q)$. Note also that these classes possess the following property:

Theorem 2.4
If $f_1 \in C_{L_1}^{k,p}(Q)$, $f_2 \in C_{L_2}^{k,p}(Q)$ and $\alpha_1,\ \alpha_2 \in \mathbb{R}$, then for

$$L_3 = |\alpha_1|L_1 + |\alpha_2|L_2$$

we have $\alpha_1 f_1 + \alpha_2 f_2 \in C_{L_3}^{k,p}(Q)$.

♢

One of the most important classes of differentiable functions is $C_L^{1,1}(\mathbb{R}^n)$, *the class of functions with Lipschitz continuous gradient*, which means that

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall f \in C_L^{1,1};\ \boldsymbol{x},\ \boldsymbol{y} \in \mathbb{R}^n.$$

Lemmas 2.2
A function $f(\cdot)$ belongs to the class $C_L^{2,1}(\mathbb{R}^n) \subset C_L^{1,1}(\mathbb{R}^n)$ if and only if for all $\boldsymbol{x} \in \mathbb{R}^n$ we have

$$\left\|\nabla^2 f(\boldsymbol{x})\right\| \leq L. \tag{2.7}$$

♡

Proof (of Lemma. 2.2) pass ∎

For Eq. 2.7, it can be rewritten as

$$-L\boldsymbol{I}_n \preceq \nabla^2 f(\boldsymbol{x}) \preceq L\boldsymbol{I}_n. \tag{2.8}$$

The next statement is important for the geometric interpretation of functions in $C_L^{1,1}(\mathbb{R}^n)$

Lemmas 2.3
Let $f \in C_L^{1,1}(\mathbb{R}^n)$. Then we have

$$|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle| \leq \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{2.9}$$

♡

Proof (of Lemma. 2.3) pass ∎

The second main class of functions is type $C_M^{2,2}$, *the class of twice differentiable functions with Lipschitz continuous Hessian.* Recall that for $f \in C_M^{2,2}(\mathbb{R}^n)$, we have

$$\left\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\right\| \leq M\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x},\ \boldsymbol{y} \in \mathbb{R}^n.$$

Lemmas 2.4
Let $f \in C_M^{2,2}(\mathbb{R}^n)$. The for all $\boldsymbol{x},\ \boldsymbol{y} \in \mathbb{R}^n$ we have

$$\left\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})\right\| \leq \frac{M}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{2.10}$$

$$\left|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle - \frac{1}{2}\left\langle \nabla^2 f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\right\rangle\right| \leq \frac{M}{6}\|\boldsymbol{y} - \boldsymbol{x}\|^3. \tag{2.11}$$

♡

Corollary 2.1
Let $f \in C_M^{2,2}(\mathbb{R}^n)$ and $\boldsymbol{x},\ \boldsymbol{y} \in \mathbb{R}^n$ with $\|\boldsymbol{y} - \boldsymbol{x}\| = r$. Then

$$\nabla^2 f(\boldsymbol{x}) - Mr\boldsymbol{I}_n \preceq \nabla^2 f(\boldsymbol{y}) \preceq \nabla^2 f(\boldsymbol{x}) + Mr\boldsymbol{I}_n.$$

♠

2.3 The Gradient Method

As we have already seen, the *antigradient* is the direction of locally steepest descent of a differentiable function. So let's consider the following strategy.

Gradient Method

Choose $\boldsymbol{x}_0 \in \mathbb{R}^n$.
Iterate $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h_k \nabla f(\boldsymbol{x}_k),\ k = 0, 1, \ldots$.

This scheme is called *Gradient Method*. The scalar factor h_k are called *step sizes* and they are positive.

There are many variants of this method, which differ one from another by the *step-size strategy*.

1. The sequence $\{h_k\}_{k=0}^{\infty}$ is chosen *in advance*. For instance

$$h_k = h > 0, \text{ (constant step)} \quad h_k = \frac{h}{\sqrt{k+1}}$$

   This is the simplest one. It is often used in the context of Convex Optimization. In this framework, the behavior of functions is much predictable than in general nonlinear case.

2. *Full relaxation*:

$$h_k = \arg\min_{h \geq 0} f(\boldsymbol{x}_k - h \nabla f(\boldsymbol{x}_k)).$$

   This strategy is completely *theoritical*. It is never used in practice since even in one-dimensional case we cannot find the *exact minimum in finite time*.

3. The *Armijo* rule: Find $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h \nabla f(\boldsymbol{x}_k)$ with $h > 0$ such that

$$\begin{aligned} \alpha \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle &\leq f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \\ \beta \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k+1} \rangle &\geq f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \end{aligned}$$

   where $0 < \alpha < \beta < 1$ are some fixed parameters.

The third strategy is used in the majority of practical algorithms. It has the following geometric interpretation. Let us fix $\boldsymbol{x} \in \mathbb{R}^n$ assuming that $\nabla f(\boldsymbol{x}) \neq 0$. Consider the following function of one variable:

$$\phi(h) = f(\boldsymbol{x} - h \nabla f(\boldsymbol{x})), \quad h \geq 0.$$

Then the step-size values acceptable for this strategy belong to thee part of the graph of ϕ which is located between 2 linear functions:

$$\phi_1(h) = f(\boldsymbol{x}) - \alpha h \|\nabla f(\boldsymbol{x})\|^2, \quad \phi_2(h) = f(\boldsymbol{x}) - \beta h \|\nabla f(\boldsymbol{x})\|^2.$$

Note that $\phi(0) = \phi_1(0) = \phi_2(0)$ and $\phi'(0) < \phi_2'(0) < \phi_1'(0) < 0$. Therefore, the acceptable values exist unless $\phi(\cdot)$ is not bounded below.

Let us estimate the performace of Gradient Method. Consider the problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}), \tag{2.12}$$

with $f \in C_L^{1,1}(\mathbb{R}^n)$, and assume that $f(\cdot)$ is bounded below on $\mathbb{R}^n$.

For one gradient step, consider $\boldsymbol{y} = \boldsymbol{x} - h \nabla f(\boldsymbol{x})$. Then, in view of Lemma. 2.3, we have

$$\begin{aligned} f(\boldsymbol{y}) &\leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2 \\ &= f(\boldsymbol{x}) - h \left(1 - \frac{h}{2} L\right) \|\nabla f(\boldsymbol{x})\|^2. \end{aligned} \tag{2.13}$$

In order to get the best upper bound for the possible decrease of the objective function, we have to solve the following one-dimensional problem:

$$\Delta(h) = -h\left(1 - \frac{h}{2}L\right) \to \min_h.$$

Computing the derivative of this function, we conclude that the optimal step size $h^* = \frac{1}{L}$, which is a minimum of $\Delta(h)$ since $\Delta''(h) = L > 0$.

Therefore, we can get that one step of the Gradient Method decreases the value of the objective function at least as follows:

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) - \frac{1}{2L}\|\nabla f(\boldsymbol{x})\|^2.$$

Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h_k \nabla f(\boldsymbol{x}_k)$.

1. **For the constant step strategy**: $h_k = h$. Then we have

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq h\left(1 - \frac{1}{2}Lh\right)\|\nabla f(\boldsymbol{x}_k)\|^2.$$

   If we choose $h_k = \frac{2\alpha}{L}$ with $\alpha \in (0, 1)$, then

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq \frac{2}{L}\alpha(1-\alpha)\|\nabla f(\boldsymbol{x}_k)\|^2.$$

2. **For the full relaxation strategy**:

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq \frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2$$

3. **For the Armijo rule**:

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \leq \beta\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k+1}\rangle = \beta h_k\|\nabla f(\boldsymbol{x}_k)\|^2.$$

   From 2.13, we obtain

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq h_k\left(1 - \frac{h_k}{2}L\right)\|\nabla f(\boldsymbol{x}_k)\|^2.$$

   Thus, $h_k \geq \frac{2}{L}(1-\beta)$. Furthermore,

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq \alpha\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k+1}\rangle = \alpha h_k\|\nabla f(\boldsymbol{x}_k)\|^2.$$

   '' Combining this inequality with the previous one, we conclude that

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq \frac{2}{L}\alpha(1-\beta)\|\nabla f(\boldsymbol{x}_k)\|^2.$$

Above all, we have proved that we have

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1}) \geq \frac{\omega}{L}\|\nabla f(\boldsymbol{x}_k)\|^2, \tag{2.14}$$

where ω is some positive constant.

Now we are already to estimate the performace of Gradient Method. Summing up the inequalities

2.14 for $k = 0, \ldots, N$, we obtain

$$\frac{\omega}{L} \sum_{k=0}^{N} \|\nabla f(\boldsymbol{x}_k)\|^2 \leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{N+1}) \leq f(\boldsymbol{x}_0) - f^*, \tag{2.15}$$

where f^* is lower bounds for the values of objective function in the problem 2.12. As a simple consequence of bound 2.15, we have

$$\lim_{k \to \infty} \|\nabla f(\boldsymbol{x}_k)\| = 0$$

However, we also say something about the *rate of convergence*. We define

$$g_N^* = \min_{0 \leq k \leq N} \|\nabla f(\boldsymbol{x}_k)\|.$$

Then in view of 2.15, we come to the following inequality:

$$g_N^* \leq \frac{1}{N+1} \left[\frac{L}{\omega} (f(\boldsymbol{x}_0) - f^*) \right]^{1/2} \tag{2.16}$$

The right-hand side of this inequality describes the rate of convergence of the sequence of the sequence $\{g_N^*\}$ to zero. Our current goal is to approach a minimum of the optimization problem 2.12. However, in general, even this goal is unreachable for the Gradient Method.

EXAMPLE 2.1 Consider the following function of two variables:

$$f(\boldsymbol{x}) \equiv f(x^{(1)}, x^{(2)}) = \frac{1}{2} \left(x^{(1)} \right)^2 + \frac{1}{4} \left(x^{(2)} \right)^4 - \frac{1}{2} \left(x^{(2)} \right)^2.$$

The gradient of this function is $\nabla f(\boldsymbol{x}) = \left(x^{(1)}, \left(x^{(2)} \right)^3 - x^{(2)} \right)^\top$. There are only three points which can pretend to be a local minimum of this function:

$$\boldsymbol{x}_1^* = (0, 0), \quad \boldsymbol{x}_2^* = (0, -1), \quad \boldsymbol{x}_3^* = (0, 1).$$

Computing the Hessian of this function,

$$\nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 3 \left(x^{(2)} \right)^2 - 1 \end{pmatrix},$$

we conclude that $\boldsymbol{x}_2^*$ and $\boldsymbol{x}_3^*$ are isolated local minima, but $\boldsymbol{x}_1^*$ is only a *stationary point* of this function. In deed, $f(\boldsymbol{x}_1^*) = 0$ and $f(\boldsymbol{x}_1^* + \epsilon \boldsymbol{e}_2) = \frac{\epsilon^4}{4} - \frac{\epsilon^2}{2} < 0$ for ϵ small enough.

Let's consider now trajectory of the Gradient Method which starts at $\boldsymbol{x}_0 = (1, 0)$. The entire sequence will have second coordinate equal to zero. This means that this sequence converges to $\boldsymbol{x}_1^*$.$*$

The *rate of convergence* delivers an *upper* complexity bound for the corresponding problem class. Such a bound is always justified by some numerical method. A method for which the upper complexity bound is proportional to the *lower* complexity bound of the problem class is said to be *optimal*.

Consider the following problem class $\mathscr{G}_*$.

Problem class $\mathscr{G}_*$

- **Model**:
  1. Unconstrained minimization.
  2. $f \in C^{1,1}_L(\mathbb{R}^n)$.
  3. $f(\cdot)$ is bounded below by the value f^*.
- **Oracle**: First-Order Black Box
- ε**-solution**: $f(\bar{\boldsymbol{x}}) \leq f(\boldsymbol{x}_0),\ \|\nabla f(\bar{\boldsymbol{x}})\| \leq \epsilon$

Note that the inequality 2.16 can be used in order to obtain an upper bound for the number of steps (= calls of oracle), which is necessary to find a point where the norm of the gradient is small. For that, let us write down the following inequality:

$$g^*_N \leq \frac{1}{N+1}\left[\frac{L}{\omega}(f(\boldsymbol{x}_0) - f^*)\right]^{1/2} \leq \epsilon. \tag{2.17}$$

Therefore, if $N + 1 \geq \frac{L}{\omega\epsilon^2}(f(\boldsymbol{x}_0) - f^*)$, then we necessarily have $g^*_N \leq \epsilon$. The lower complexity bound for the class $\mathscr{G}_*$ is unknown.

Theorem 2.5
Let the function $f(\cdot)$ satisfy our assumptions and let the starting point $\boldsymbol{x}_0$ be close enough to a strict local minimum $\boldsymbol{x}^*$:

$$r_0 = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\| < \bar{r} = \frac{2\mu}{M}.$$

Then the Gradient Method with step size converges as follows:

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq \frac{\bar{r}r_0}{\bar{r} - r_0}\left(1 - \frac{2\mu}{L + 3\mu}\right)^k. \qquad \diamondsuit$$

This type of rate of convergence is called *linear*.

2.4 Newton's Method

Newton's Method is widely known as a technique for finding a root of univariate function. Let $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$. Consider the equation

$$\phi(t^*) = 0.$$

Assume that we know some $t \in \mathbb{R}$ which is close enough to t^*. Note that

$$\phi(t + \Delta t) = \phi(t) + \phi'(t)\Delta t + o(|\Delta t|).$$

Therefore, the solution can be approximated by the solution of the following *linear* equation:

$$\phi(t) + \phi'(t)\Delta t = 0.$$

Under some conditions, we can expect the displacement Δt to be a good approximation to the optimal displacement $\Delta t^* = t^* - t$. Converting this idea into an algorithm, we get the process

$$t_{k+1} = t_k - \frac{\phi(t_k)}{\phi'(t_k)}.$$

This scheme can be naturally extended to the problem of finding a solution to a system of nonlinear equations

$$\boldsymbol{F}(\boldsymbol{x}) = 0,\ \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{F}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n.$$

In this case, the scheme is as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\boldsymbol{F}'(\boldsymbol{x}_k)]^{-1}\boldsymbol{F}(\boldsymbol{x}_k).$$

Finally, in view of Theorem. 2.1, we can replace th unconstrained minimization problem 2.12 by the problem of *finding a root of the nonlinear system*

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

This replacement is not completely equivalent, but it works in *nondegenerate* situations. In this case, the Newton system is as follows:

$$\nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x})\Delta\boldsymbol{x} = 0.$$

Hence, the Newton's Method for optimization problems can be written in the following form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k).$$

Newton's Method from Quadratic Approximation

Consider the approximation

$$\phi(\boldsymbol{x}) = f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2}\left\langle \nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \right\rangle.$$

Assume that $\nabla^2 f(\boldsymbol{x}_k) \succ 0$. Then we can choose $\boldsymbol{x}_{k+1}$ as the minimizer of the quadratic function $\phi(\cdot)$, This means that

$$\nabla\phi(\boldsymbol{x}_{k+1}) = \nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) = 0,$$

and we come again to Newton's process.

The convergence of the Newton's Method in a neighborhood of a strict local minimum is very fast. Nevertheless, this method has two serious drawbacks:

1. It can break down if $\nabla^2 f(\boldsymbol{x}_k)$ is degenerate.
2. Newton's process can diverge.

EXAMPLE 2.2 Consider following univariate function

$$\phi(t) = \frac{t}{\sqrt{1+t^2}}.$$

Clearly, $t^* = 0$. Note that

$$\phi'(t) = \frac{1}{(1+t^2)^{3/2}}.$$

Therefore Newton's process is as follows:

$$t_{k+1} = t_k - \frac{\phi(t_k)}{\phi'(t_k)} = t_k - \frac{t_k}{\sqrt{1+t_k^2}} \cdot [1+t_k^2]^{3/2} = -t_k^3.$$

Thus, if $|t_0| < 1$, then this method converges and the convergence is extremely fast. The points ± 1 are oscillation points of this scheme. If $|t|_0 > 1$, then this method diverges. $*$

In order to avoid a possible diverge, in practice we can apply the *damped Newton's method*:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h_k[\nabla^2 f(\boldsymbol{x}_k)]^{-1}\nabla f(\boldsymbol{x}_k),$$

where $h_k > 0$ is a step size parameter. At the initial stage of the method we can use the same step size strategies as for the gradient scheme. At the final stage, it is reasonable to choose $h_k = 1$. Another possibility for ensuring the global convergence of the scheme consists in using *Cubic Regularization*, This approach will be studied in the later chapters.

Theorem 2.6
Let the function $f(\cdot)$ satisfy our assumptions. Suppose that the initial starting point $\boldsymbol{x}_0$ is close enough to $\boldsymbol{x}^*$:

$$\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq \bar{r} = \frac{2\mu}{3M}.$$

Then $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq \bar{r}$ for all k and the Newton's Method converges quadratically:

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \frac{M\|\boldsymbol{x}_k - \boldsymbol{x}^*\|}{2(\mu - M\|\boldsymbol{x}_k - \boldsymbol{x}^*\|)}.$$ $\diamondsuit$

1. **Sublinear rate**
2. **Linear rate**
3. **Quadratic rate**

3 First-Order Methods in Nonlinear Optimization

3.1 The Gradient Method and Newton's Method: What is Different?

For the optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

with $f \in C_M^{2,2}(\mathbb{R}^n)$.

1. **Gradient Method**:
   - $\boldsymbol{x}_{k+1} = \boldsymbol{x} - h_k \nabla f(\boldsymbol{x}_k),\ h_k > 0.$
   - Local rate of convergence: linear rate.
   - The search direction: the *antigradient*.
2. **Newton Method**:
   - $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k).$
   - Local rate of convergence: Quadratic rate
   - The search direction: multiply the antigradient by *inverse Hessian*.

Let us try to derive these directions using some "universal" reasoning.

Fix a point $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ and consider the following approximation of the function $f(\cdot)$:

$$\phi_1(\boldsymbol{x}) = f(\bar{\boldsymbol{x}}) + \langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle + \frac{1}{2h} \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|^2,\ h > 0.$$

According to the first-order condition, we obtain the following equation for $\boldsymbol{x}_1^*$, the unconstrained minimum of this function

$$\nabla \phi_1(\boldsymbol{x}_1^*) = \nabla f(\bar{\boldsymbol{x}}) + \frac{1}{h}(\boldsymbol{x}_1^* - \bar{\boldsymbol{x}}) = 0.$$

Thus, $\boldsymbol{x}_1^* = \bar{\boldsymbol{x}} - h \nabla f(\bar{\boldsymbol{x}})$. According to Lemma. 2.3 if $h \in (0, \frac{1}{L}]$, then the function ϕ_1 is a *global upper* approximation of $f(\cdot)$:

$$f(\boldsymbol{x}) \leq \phi_1(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^n.$$

This fact is responsible for the global convergence of the Gradient Method.

Futher, consider a quadratic approximation of the function $f(\cdot)$:

$$\phi_2(\boldsymbol{x}) = f(\bar{\boldsymbol{x}}) + \langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle + \frac{1}{2} \left\langle \nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}}), \boldsymbol{x} - \bar{\boldsymbol{x}} \right\rangle.$$

We have already seen that the minimum of this function is

$$\boldsymbol{x}_2^* = \bar{\boldsymbol{x}} - [\nabla^2 f(\bar{\boldsymbol{x}})]^{-1} \nabla f(\bar{\boldsymbol{x}}),$$

and this is exactly the iterate of the Newton's Method.

Therefore, we try to use some quadratic approximations of function $f(\cdot)$, which are better than ϕ_1 and which are less expensive than ϕ_2.

Let G be a symmetric positive definite $n \times n$-matrix. Define

$$\phi_G(\boldsymbol{x}) = f(\bar{\boldsymbol{x}}) + \langle \nabla f(\bar{\boldsymbol{x}}), \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle + \frac{1}{2} \left\langle \boldsymbol{G}(\boldsymbol{x} - \bar{\boldsymbol{x}}), \boldsymbol{x} - \bar{\boldsymbol{x}} \right\rangle.$$

Computing the minimizer of $\phi_G(\cdot)$ from the equation

$$\nabla\phi_G(\boldsymbol{x}_G^*) = \nabla f(\bar{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}_G^* - \bar{\boldsymbol{x}}) = 0.$$

we obtain

$$\boldsymbol{x}_G^* = \bar{\boldsymbol{x}} - \boldsymbol{G}^{-1}\nabla f(\bar{\boldsymbol{x}}). \tag{3.1}$$

Definition 3.1 (Variable Metric Methods)
The first-order methods, which form a sequence of matrices

$$\{\boldsymbol{G}_k\}: \ \boldsymbol{G}_k \to \nabla^2 f(\boldsymbol{x}^*) \text{ or } \{\boldsymbol{H}_k\} \equiv \boldsymbol{G}_k^{-1} \to \nabla^2 f(\boldsymbol{x}^*)$$

are called variable metric *methods or* quasi-Newton *methods.* ♣

In these methods, only the gradients are involved in the process of generating the sequences $\{\boldsymbol{G}_k\}$ or $\{\boldsymbol{H}_k\}$. Let us provide updating rule 3.1 one more interpretation.

Consider a symmetric positive definite $n \times n$-matrix $\boldsymbol{A}$. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ define

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{A}} = \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y} \rangle, \quad \|\boldsymbol{x}\|_{\boldsymbol{A}} = \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x} \rangle^{1/2}.$$

This function $\|\cdot\|_{\boldsymbol{A}}$ is treated as a new norm on $\mathbb{R}^n$. Note that the topologically this new norm is equivalent to the old one

$$\lambda_{\min}(\boldsymbol{A})^{1/2}\|\boldsymbol{x}\| \le \|\boldsymbol{x}\|_{\boldsymbol{A}} \le \lambda_{\max}(\boldsymbol{A})^{1/2}\|\boldsymbol{x}\|,$$

where $\lambda_{\min}(\boldsymbol{A})$ and $\lambda_{\max}(\boldsymbol{A})$ are the smallest and largest eigenvalues of the matrix $\boldsymbol{A}$. Therefore we can obtain

$$\begin{aligned} f(\boldsymbol{x}+\boldsymbol{h}) &= f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{h} \rangle + \frac{1}{2}\left\langle \nabla^2 f(\boldsymbol{x})\boldsymbol{h}, \boldsymbol{h} \right\rangle + o(\|\boldsymbol{h}\|) \\ &= f(\boldsymbol{x}) + \left\langle \boldsymbol{A}^{-1}\nabla f(\boldsymbol{x}), \boldsymbol{h} \right\rangle_{\boldsymbol{A}} + \frac{1}{2}\left\langle \boldsymbol{A}^{-1}\nabla^2 f(\boldsymbol{x})\boldsymbol{h}, \boldsymbol{h} \right\rangle_{\boldsymbol{A}} + o(\|\boldsymbol{h}\|_{\boldsymbol{A}}) \end{aligned}$$

Hence, $\nabla f_{\boldsymbol{A}}(\boldsymbol{x}) = \boldsymbol{A}^{-1}\nabla f(\boldsymbol{x})$ is the new gradient and $\nabla^2 f_{\boldsymbol{A}}(\boldsymbol{x}) = \boldsymbol{A}^{-1}\nabla^2 f(\boldsymbol{x})$ is the new Hessian.

Algorithm 3: Variable Metric Method

Input : Starting point $\boldsymbol{x}_0$ and accuracy $\epsilon > 0$.
Output: Local minimum $\boldsymbol{x}^*$ and $f(\boldsymbol{x}^*)$.
Initialization: Set $\boldsymbol{H}_0 = \boldsymbol{I}_n$ and $k = 0$. Compute $f(\boldsymbol{x}_0)$ and $\nabla f(\boldsymbol{x}_0)$.
while *not* $(\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon$ *or* $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| \le \epsilon$ *or* $\|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)\| \le \epsilon)$ **do**
  Set $\boldsymbol{p} = \boldsymbol{H}_k\nabla f(\boldsymbol{x}_k)$.
  Find $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h_k\boldsymbol{p}_k$.
  Compute $f(\boldsymbol{x}_{k+1})$ and $\nabla f(\boldsymbol{x}_{k+1})$.
  Update the matrix $\boldsymbol{H}_k$ to $\boldsymbol{H}_{k+1}$.
end

Quasi-Newton Method

Choose $\boldsymbol{H}_{k+1} = \boldsymbol{H}_{k+1}^\top \succ 0$ such that

$$\boldsymbol{H}_{k+1}(\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)) = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k.$$

Define

$$\Delta \boldsymbol{H}_k = \boldsymbol{H}_{k+1} - \boldsymbol{H}_k, \quad , \boldsymbol{\gamma}_k = \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k), \quad \boldsymbol{\delta}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k.$$

Then the quasi-Newton relation is satisfied by the following updating rules.

1. Rank-one correction scheme:

$$\Delta \boldsymbol{H}_k = \frac{(\boldsymbol{\delta}_k - \boldsymbol{H}_k\boldsymbol{\gamma}_k)(\boldsymbol{\delta}_k - \boldsymbol{H}_k\boldsymbol{\gamma}_k)^\top}{\langle \boldsymbol{\delta}_k - \boldsymbol{H}_k\boldsymbol{\gamma}_k, \boldsymbol{\gamma}_k\rangle}$$

2. Davidon–Fletcher–Powell scheme (DFP):

$$\Delta \boldsymbol{H}_k = \frac{\boldsymbol{\delta}_k\boldsymbol{\delta}_k^\top}{\langle \boldsymbol{\gamma}_k, \boldsymbol{\delta}_k\rangle} - \frac{\boldsymbol{H}_k\boldsymbol{\gamma}_k\boldsymbol{\gamma}_k^\top\boldsymbol{H}_k}{\langle \boldsymbol{H}_k\boldsymbol{\gamma}_k, \boldsymbol{\gamma}_k\rangle}$$

3. Broyden–Fletcher–Goldfarb–Shanno scheme (BFGS):

$$\Delta \boldsymbol{H}_k = \beta_k \frac{\boldsymbol{\delta}_k\boldsymbol{\delta}_k^\top}{\langle \boldsymbol{\gamma}_k, \boldsymbol{\delta}_k\rangle} - \frac{\boldsymbol{H}_k\boldsymbol{\gamma}_k\boldsymbol{\delta}_k^\top + \boldsymbol{\delta}_k\boldsymbol{\gamma}_k^\top\boldsymbol{H}_k}{\langle \boldsymbol{H}_k\boldsymbol{\gamma}_k, \boldsymbol{\gamma}_k\rangle}$$

where $\beta_k = 1 + \langle \boldsymbol{H}_k\boldsymbol{\gamma}_k, \boldsymbol{\gamma}_k, \boldsymbol{\delta}_k\rangle$

From the computational point of view, BFGS is considered to be the most stable scheme.

For variable metric methods, in a neighborhood of a strict local minimum $\boldsymbol{x}^*$ they demonstrate a *superlinear* rate of convergence: for any $\boldsymbol{x}^0 \in \mathbb{R}^n$ close enough to $\boldsymbol{x}^*$ there exists a number N such that for all $k \geq N$ we have

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \text{const} \cdot \|\boldsymbol{x}_k - \boldsymbol{x}^*\| \cdot \|\boldsymbol{x}_{k-n} - \boldsymbol{x}^*\|.$$

As far as the worst-case global convergence is concerned, these methods are not better than the Gradient Method.

In the variable metric schemes it is necessary to store and update a symmetric $n \times n$-matrix. Thus, each iteration needs $O(n^2)$ auxiliary arithmetic operations. This feature is considered as one of the main drawbacks of the variable metric methods.

3.2 Conjugate Gradients

Consider the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \tag{3.2}$$

with $f(\boldsymbol{x}) = \alpha + \langle \boldsymbol{a}, \boldsymbol{x}\rangle + \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle$ and $\boldsymbol{A} = \boldsymbol{A}^\top \succ 0$. The solution of this problem is $\boldsymbol{x}^* = -\boldsymbol{A}^{-1}\boldsymbol{a}$. Therefore, our objective function can be written in the following form:

$$\begin{aligned} f(\boldsymbol{x}) &= \alpha + \langle \boldsymbol{a}, \boldsymbol{x}\rangle + \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle = \alpha - \langle \boldsymbol{A}\boldsymbol{x}^*, \boldsymbol{x}\rangle + \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle \\ &= \alpha - \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}^*, \boldsymbol{x}^*\rangle + \frac{1}{2}\langle \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*), \boldsymbol{x} - \boldsymbol{x}^*\rangle \end{aligned}$$

Thus, $f^* = \alpha - \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{x}^*, \boldsymbol{x}^*\rangle$ and $\nabla f(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*)$.

Algorithm 4: Conjugate Gradient Method

Input : Starting point $\boldsymbol{x}_0$ and accuracy $\epsilon > 0$.
Output: Local minimum $\boldsymbol{x}^*$ and $f(\boldsymbol{x}^*)$.
Initialization: Set $k = 0$. Compute $f(\boldsymbol{x}_0)$ and $\nabla f(\boldsymbol{x}_0)$. Set $\boldsymbol{p}_0 = \nabla f(\boldsymbol{x}_0)$.
while *not (*$\|\nabla f(\boldsymbol{x}_k)\| \leq \epsilon$ *or* $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| \leq \epsilon$ *or* $\|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)\| \leq \epsilon$*)* **do**
  Find $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h_k \boldsymbol{p}_k$.
  Compute coeffcient β_k.
  Define $\boldsymbol{p}_{k+1} = \nabla f(\boldsymbol{x}_{k+1}) - \beta_k \boldsymbol{p}_k$.
end

1. Dai-Yuan:
$$\beta_k = \frac{\|\nabla f(\boldsymbol{x}_{k+1})\|}{\langle \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k\rangle}$$

2. Fletcher-Rieves:
$$\beta_k = \frac{\|\nabla f(\boldsymbol{x}_{k+1})\|}{\|\nabla f(\boldsymbol{x}_k)\|}$$

3. Polak-Ribbiere:
$$\beta_k = -\frac{\langle \nabla f(\boldsymbol{x}_{k+1}), \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)\rangle}{\|\nabla f(\boldsymbol{x}_k)\|}$$