Chapter 2: Unconstrained Optimization

1 Unconstrained optimization in one dimension

The simple unconstrained optimization problem in one dimension is: given a function $f : \mathbb{R} \to \mathbb{R}$, find $x^* \in \mathbb{R}$, such that the function $f(x)$ achieves its minimum value at $x = x^*$, i.e.,

$$f(x^*) = \min_{x \in \mathbb{R}} f(x),$$

or, equivalently,

$$x^* = \arg\min_{x \in \mathbb{R}} f(x).$$

We call $x^* \in \mathbb{R}$ the global minimum of the function $f(x)$, if $f(x^*) \leq f(x)$, for all $x \in \mathbb{R}$. We call x^* a local minimum of $f(x)$, if $f(x^*) \leq f(x)$, for all x in a neighborhood of x^*. Certainly, the global minimum of $f(x)$ is one of its local minimum. However, given a local minimum of $f(x)$, typically, it is hard to determine if it is a global minimum of the function, unless the function $f(x)$ has some special properties. For example, if the function $f(x)$ is convex, i.e., for any $x, y \in R$, and any $\alpha \in [0, 1]$,

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

then a local minimum of $f(x)$ is also the global minimum.

In general there is no algorithm to find the global minimum of a function. Most algorithms are derived by using the local approximations of the function and as a result what can be achieved numerically is to find a local minimum of the function. Also there is no exact formula in general to find a local minimum of a function and the algorithm has to be iterative in nature.

Let us first review the Taylor expansion which will be used very frequently.

Theorem 1 (Taylor's Theorem in one dimension) *Let $f(x) : \mathbb{R} \to \mathbb{R}$, be a smooth function. Given $h \in \mathbb{R}$, the expansion of $f(x + h)$ at x is*

$$f(x + h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \cdots + \frac{h^n}{n!}f^{(n)}(x) + \ldots,$$

which is also called the Taylor series.

The Taylor series can be regarded as a sequence of approximation to $f(x + h)$ with increasing accuracy. The error in approximation of $f(x + h)$ by a few terms of its Taylor series is called *truncation error*. In the above Taylor series,

$$f(x + h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \cdots + \frac{h^n}{n!}f^{(n)}(x) + \frac{h^{n+1}}{(n+1)!}f^{(n+1)}(x + \alpha h),$$

for a certain $\alpha \in [0,1]$. The truncation error in this case is

$$\frac{h^{n+1}}{(n+1)!} f^{(n+1)}(x+\alpha h),$$

which is of the order $O(h^{n+1})$ for smooth function $f(x)$.

For example, taking $n = 0$, we have

$$f(x+h) \;=\; f(x) + hf'(x+\alpha h) \;=\; f(x) + O(h).$$

Here $f(x+h)$ is approximated by the constant value $f(x)$, and the truncation error of using this constant approximation is in the order of $O(h)$. Taking $n = 1$, we have

$$f(x+h) \;=\; f(x) + hf'(x) + \frac{h^2}{2} f''(x+\alpha h) \;=\; f(x) + hf'(x) + O(h^2).$$

We call $f(x) + hf'(x)$ a linear approximation of $f(x+h)$ around x. The truncation error of this linear approximation is in the order of $O(h^2)$, which is better than the constant approximation. Taking $n = 2$,

$$f(x+h) \;=\; f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\frac{h^3}{3!}f^{(3)}(x+\alpha h) \;=\; f(x)+hf'(x)+\frac{h^2}{2}f''(x)+O(h^3),$$

we have the quadratic approximation of $f(x+h)$, with truncation error in the order of $O(h^3)$.

The following theorems are some necessary and sufficient conditions for local minimum.

Theorem 2 *Let $f(x) : \mathbb{R} \to \mathbb{R}$, be continuously differentiable. For any $x \in \mathbb{R}$, if $f'(x) \neq 0$, then there exists $h \in \mathbb{R}$, such that for all $\tau \in (0,1]$*

$$f(x+\tau h) < f(x),$$

i.e., x can not be a minimum of $f(x)$.

Proof: We know from Theorem 1, for the case of constant approximation, that

$$f(x+\tau h) \;=\; f(x) + \tau h f'(x+\alpha\tau h),$$

for a certain $\alpha \in [0,1]$.

Since $f'(x) \neq 0$, let us assume that $f'(x) < 0$; the same argument also applies to the case $f'(x) > 0$ with just a minor modification as mentioned below. Since $f'(x)$ is continuous, we can see that there exists $h \in \mathbb{R}$ (as long as $|h|$ sufficiently small, no matter positive or negative), such that for all $\tau \in (0,1]$ and $\alpha \in [0,1]$,

$$f'(x+\alpha\tau h) < 0.$$

Let such a h be positive (negative if $f'(x) > 0$), then for this h and all $\tau \in (0, 1]$

$$f(x + \tau h) \;=\; f(x) + \tau h f'(x + \alpha\tau h) \;<\; f(x),$$

i.e., x can not be a minimum of $f(x)$. $\square$

Then the following theorem immediately follows.

Theorem 3 (A necessary condition) *Let $f(x)$ be continuously differentiable. If x^* is a local minimum of $f(x)$, then $f'(x^*) = 0$.*

We call x a *critical point* of $f(x)$, if $f'(x) = 0$. Theorem 2 tells us that a local minimum of a function must be a critical point of that function.

The following theorem provides a sufficient condition for the local minimum.

Theorem 4 (A sufficient condition) *Let $f(x)$ be continuously differentiable. If there exists a point x, where $f'(x) = 0$, and $f''(x) > 0$, then $f(x)$ achieves a local minimum at x. If $f'(x) = 0$, and $f''(x) < 0$, then $f(x)$ achieves a local maximum at x.*

Proof: Let us give the proof for the case $f'(x) = 0$ and $f''(x) > 0$. From the Taylor series, for any h,

$$f(x + h) \;=\; f(x) + hf'(x) + \frac{h^2}{2} f''(x + \alpha h) \;=\; f(x) + \frac{h^2}{2} f''(x + \alpha h),$$

for a certain $\alpha \in [0, 1]$. Since $f''(x) > 0$ and $f''(x)$ is continuous, we can see that for all h small enough, all $f''(x + \alpha h) > 0$, for all $\alpha \in [0, 1]$. Therefore for any h sufficiently small,

$$f(x + h) \;=\; f(x) + \frac{h^2}{2} f''(x + \alpha h) \;>\; f(x),$$

which implies that $f(x)$ is a local minimum of the function $f(x)$. $\square$

We also have the following theorem.

Theorem 5 *Let $f(x)$ be a continuous function. If there exists a point x^*, such that $f(x)$ is strictly decreasing at a left side neighborhood of x^*, and strictly increasing at a right side neighborhood of x^*, then x^* is a local minimum of $f(x)$.*

2 Algorithms for one dimensional optimization problem

The algorithms for solving minimization problems are essentially finding critical points of $f(x)$, i.e., solving $f'(x) = 0$.

2.1 Golden section method

The interval bisection method can certainly be used to solve $f'(x) = 0$, as long as on the initial interval $[a, b]$, it satisfies that $f'(a) * f'(b) < 0$. It needs to evaluate the derivative of the function at each step.

A more sophisticated method is called the Golden Section Search method. It can only be applied to *unimodal* functions. We say that a function $f(x)$ is *unimodal* on the interval $[a, b]$, if there is a unique $x^* \in [a, b]$ which is the minimum of the function $f(x)$ on $[a, b]$ and $f(x)$ is strictly decreasing on the left of x^*, and strictly increasing at the right of x^*.

The Golden Section Search method does not need to evaluate the derivative of $f(x)$ to solve $f'(x) = 0$. Let $[a, b]$ be an interval on which the function $f(x)$ is unimodal. Choose two points x_1 and x_2 inside the interval, i.e., $a < x_1 < x_2 < b$. Evaluate the function values at x_1 and x_2. If $f(x_1) > f(x_2)$, then the minimum must be to the right of x_1, i.e., inside the interval $[x_1, b]$, and we take $[x_1, b]$ as the new interval $[a, b]$ to continue for the next step. Otherwise, If $f(x_1) < f(x_2)$, the minimum must be to the left of x_2, i.e., inside the interval $[a, x_2]$, and we take $[a, x_2]$ as the new interval to continue. We can choose the two point x_1 and x_2 to achieve a constant reduction of the interval length, and the convergence rate of the golden section method can be made linear, e.g., $x_1 = a + (b - a) * 0.382$ and $x_2 = a + 0.618 * (b - a)$.

2.2 Newton's method

The Golden section method is guaranteed to converge, as long as the function is unimodal on the initial interval. However the convergence rate is only linear, which can be very slow. The Newton's method for optimization is a method of quadratic convergence, just as the Newton's method for solving nonlinear equations.

The Newton's iterate for the minimization of $f(x)$ is simply the Newton's iterate for solving $f'(x) = 0$. When it converges, it converges to a critical value of the function $f(x)$.

Algorithm: Newton's iteration for solving $f'(x) = 0$

Given initial guess x_0
for $k = 0, 1, 2, \ldots$ **until convergence**

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

end

The Newton's iterate can also be obtained from another perspective. Given the current iterate x_k, let us determine the quadratic approximation $Q(x)$ of the function $f(x)$ in a neighborhood of x_k, which satisfies, $Q(x_k) = f(x_k)$, $Q'(x_k) = f'(x_k)$, and $Q''(x_k) = f''(x_k)$. We can see that such a quadratic approximation is determined by

$$Q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(x_k)}{2}(x - x_k)^2.$$

Thus the critical value of $Q(x)$ is achieved at $x_k - f'(x_k)/f''(x_k)$, which is a minimum if $f''(x_k)$ is positive and a maximum if $f''(x_k)$ is negative.

We know that there are some issues on the performance of the Newton's method. First of all, when the above Newton's iteration converges, it converges to a critical point of the function $f(x)$, which may not necessarily be a minimum of $f(x)$. Second, the quadratic rate of convergence only holds when the initial point is sufficiently close to a critical point of $f(x)$. When the initial point is away from a critial point, it will typically take more iterations. Third, the correction on x_k, determined by $\dfrac{f'(x_k)}{f''(x_k)}$, might be undefined or too large, when the second derivative is (or close to) zero.

To overcome such shortcomings of the Newton's method to find a minimum of the function $f(x)$, we can consider to combine it with the globally convergent Golden Section Search method. However, the Golden Section Search method can only be used for functions of unimode, which is not general at all.

Another possibility is to apply the similar approach for solving the nonlinear equation $f'(x) = 0$, by combining the Newton's method with the interval bisection method. It will become globally convergent. However, as mentioned earlier, it just converges to a critical point of $f(x)$, not necessarily, a minimum of $f(x)$.

In the following, we discuss a *backtracking* approach in the Newton's method to find a minimum of function. The result algorithm will become globally convergent to a minimum of the function.

Given the current iterate x_k, since we are looking for a minimum point of the function $f(x)$, it is reasonable to require that the new iterate x_{k+1} satisfies $f(x_{k+1}) < f(x_k)$. If the Newton's iterate

$$x_N = x_k - \frac{f'(x_k)}{f''(x_k)}$$

satisfies $f(x_N) < f(x_k)$, then we just take $x_{k+1} = x_N$ and continue the next step. Otherwise, if $f(x_N) \geq f(x_k)$, we need to determine a new iterate x_{k+1}. As long as $f'(x_k) \neq 0$, we know from the Taylor expansion (or Theorem 2) that we can always find a point x_{k+1} close enough to x_k such that $f(x_{k+1}) < f(x_k)$. If $f'(x_k) < 0$, i.e., if the function is decreasing at x_k, then we can find such a x_{k+1} to the right of x_k; if $f'(x_k) > 0$, i.e., if the function is increasing at x_k, then we can find such a x_{k+1} to the left of x_k. At the same time, we also need that x_{k+1} is not too close to x_k, such that there is a sufficient improvement of the new iterate compared with the old one. Based on such considerations, after computing the Newton's point x_N and in this case $f(x_N) \geq f(x_k)$, we first take

$$x_{test} = x_k \pm \frac{|x_N - x_k|}{2},$$

depending on whether $f'(x_k) < 0$ or $f'(x_k) > 0$. If $f(x_{test}) < f(x_k)$, then take $x_{k+1} = x_{test}$. Otherwise, we continue to take x_{test} as the middle point of x_k and x_{test} until $f(x_{test}) < f(x_k)$ is satisfied and then take $x_{k+1} = x_{test}$. We call such a procedure the backtracking approach.

The algorithm is as following.

Algorithm: A globally convergent Newton's method for minimizing $f(x)$

Given initial guess x_0
for $k = 0, 1, 2 \ldots$**, until convergence**

$$x_N = x_k - \frac{f'(x_k)}{f''(x_k)}$$

if $f(x_N) < f(x_k)$
  $x_{k+1} = x_N$ **(take Newton iterate)**
else
  % (Implement Backtracking)
  if $f'(x_k) < 0$
    $x_{test} = x_k + \frac{|x_N - x_k|}{2}$
    while $f(x_{test}) \geq f(x_k)$
      $x_{test} = (x_k + x_{test})/2$
    end while
  else
    $x_{test} = x_k - \frac{|x_N - x_k|}{2}$
    while $f(x_{test}) \geq f(x_k)$
      $x_{test} = (x_k + x_{test})/2$
    end while
  end if
  $x_{k+1} = x_{test}$
end if
end for

---

3 Unconstrained optimization in multidimensions

Given a function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, we seek a minimum of $f(\mathbf{x})$. We need to solve the following unconstrained optimization problem: find $\mathbf{x}^* \in \mathbb{R}^n$, such that the function value $f(\mathbf{x})$ is minimized at $\mathbf{x} = \mathbf{x}^*$, i.e.,

$$\mathbf{x}^* \;=\; \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}).$$

We say $\mathbf{x}^*$ is the global minimum of $f(\mathbf{x})$, if $f(\mathbf{x}^*) \leq f(\mathbf{x})$, for all $\mathbf{x} \in \mathbb{R}^n$. $\mathbf{x}^*$ is a local minimum of $f(\mathbf{x})$, if $f(\mathbf{x}^*) \leq f(\mathbf{x})$, for all $\mathbf{x}$ in a neighborhood of $\mathbf{x}^*$. The local minimum is *strict*, if $f(\mathbf{x}^*) < f(\mathbf{x})$, for all $\mathbf{x}$ in a neighborhood of $\mathbf{x}^*$.

One example of the unconstrained optimization problem is the linear least squares problem. Given an $m \times n$ matrix A, where we assume that $m > n$ and the rank of A is n, and an m-dimensional column vector $\mathbf{b}$, we find $\mathbf{x}^* \in \mathbb{R}^n$ such that the residual $\|\mathbf{b} - A\mathbf{x}\|_2^2$

is minimized among all $\mathbf{x} \in \mathbb{R}^n$ at $\mathbf{x} = \mathbf{x}^*$. Here the residual can be written as a function of $\mathbf{x}$, i.e.,

$$f(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2^2 = (\mathbf{b} - A\mathbf{x})^T(\mathbf{b} - A\mathbf{x}) = \mathbf{b}^T\mathbf{b} - 2\mathbf{x}^T A^T\mathbf{b} + \mathbf{x}^T A^T A\mathbf{x}.$$

The linear least squares problem is then equivalent to

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

Before we discuss the algorithms for solving unconstrained optimization problems, let us first look at some definitions and theories for functions of multivariables.

3.1 Basics of Multivariable Calculus

Let $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ be a function with continuous derivatives. The gradient of $f(\mathbf{x})$ is defined as the column vector containing the first order partial derivatives of $f(\mathbf{x})$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\ \mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}.$$

The *Hessian* $\nabla^2 f(\mathbf{x})$, of $f(\mathbf{x})$, is the matrix defined by the second order partial derivatives of $f(\mathbf{x})$, as

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial^2 x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\ \mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\ \mathbf{x})}{\partial^2 x_2 \partial x_1} & \frac{\partial^2 f(\ \mathbf{x})}{\partial^2 x_2} & \cdots & \frac{\partial^2 f(\ \mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\ \mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\ \mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\ \mathbf{x})}{\partial^2 x_n} \end{bmatrix},$$

which is symmetric.

The following theorem is about the Taylor expansion for functions of multivariables.

Theorem 6 *Let $f : \mathbb{R}^n \to \mathbb{R}$, have continuous first and second partial derivatives. For any $\mathbf{x} \in \mathbb{R}^n$, and a direction $\mathbf{p} \in \mathbb{R}^n$, there exist certain positive values $\alpha, \beta \in (0, 1)$, such that*

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + \alpha\mathbf{p})^T\mathbf{p},$$
$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\nabla^2 f(\mathbf{x} + \beta\mathbf{p})\mathbf{p}.$$

Using Theorem 6, we can prove the following necessary and sufficient conditions for the local minimums of functions of multivariables.

Theorem 7 (A First-order Necessary conditions) *Let $f : \mathbb{R}^n \to \mathbb{R}$, have continuous first order partial derivatives. If $\mathbf{x}^*$ is a local minimum of $f(\mathbf{x})$, then $\nabla f(\mathbf{x}^*) = 0$.*

Proof: From Theorem 6, we know for any given vector $\mathbf{p} \in R^n$, there exists a certain $\alpha \in (0, 1)$ such that

$$f(\mathbf{x}^* + \mathbf{p}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^* + \alpha\mathbf{p})^T\mathbf{p}.$$

Let

$$\mathbf{p} = -\beta\nabla f(\mathbf{x}^*),$$

where β is any positive number.

Suppose $\nabla f(\mathbf{x}^*) \neq 0$. Since $\nabla f(\mathbf{x})$ is continuous at $\mathbf{x}^*$, we know if we take β small enough, $\mathbf{x}^* + \mathbf{p}$ will be in a neighborhood of $\mathbf{x}^*$ where for all $\mathbf{x}$ in that neighborhood $\nabla f(\mathbf{x})^T\nabla f(\mathbf{x}^*) > 0$. Take such small β in the definition of $\mathbf{p}$, then we have

$$f(\mathbf{x}^* + \mathbf{p}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^* + \alpha\mathbf{p})^T\mathbf{p} = f(\mathbf{x}^*) - \beta\nabla f(\mathbf{x}^* + \alpha\mathbf{p})^T\nabla f(\mathbf{x}^*) < f(\mathbf{x}^*).$$

Then $\mathbf{x}^*$ can not be a local minimum. Therefore $\nabla f(\mathbf{x}^*)$ must be zero. □

We call a point $\mathbf{x}^*$ a *stationary* point of $f(\mathbf{x})$, if $\nabla f(\mathbf{x}^*) = 0$. Theorem 7 tells us that a local minimum must be a stationary point. We can also see from the proof of Theorem 7 that if $\mathbf{x}^*$ is not a *stationary* point of $f(\mathbf{x})$, i.e., if $\nabla f(\mathbf{x}^*) \neq 0$, then we can always find a point moving from $\mathbf{x}^*$ along the direction $-\nabla f(\mathbf{x}^*)$ where the function has a lower value. Therefore, if $\nabla f(\mathbf{x}^*) \neq 0$, then we call $-\nabla f(\mathbf{x}^*)$ a decreasing direction at $\mathbf{x}^*$.

Before we prove a second order sufficient condition for the local minimums, let us first review the positive definiteness of a matrix. We say that a matrix A is symmetric positive definite, if

$$\mathbf{x}^T A\mathbf{x} > 0, \quad \text{for any vector } \mathbf{x} \neq \mathbf{0}.$$

Theorem 8 (A second-order sufficient conditions) *Let $f : \mathbb{R}^n \to \mathbb{R}$, have continuous first and second order partial derivatives. If $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive definite, then $\mathbf{x}^*$ is a local minimum of $f(\mathbf{x})$.*

Proof: Since $\nabla^2 f(\mathbf{x})$ is continuous and symmetric positive definite at $\mathbf{x}^*$, there exists a neighborhood of $\mathbf{x}^*$, denoted by $\mathcal{N}_{\mathbf{x}^*}$, where for each $\mathbf{x} \in \mathcal{N}_{\mathbf{x}^*}$, $\nabla^2 f(\mathbf{x})$ is also symmetric positive definite. For any $\mathbf{x} \in \mathcal{N}_{\mathbf{x}^*}$, we denote $\mathbf{x} = \mathbf{x}^* + \mathbf{p}$. Then we have

$$f(\mathbf{x}^* + \mathbf{p}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\nabla^2 f(\mathbf{x}^* + \alpha\mathbf{p})\mathbf{p} = f(\mathbf{x}^*) + \frac{1}{2}\mathbf{p}^T\nabla^2 f(\mathbf{x}^* + \alpha\mathbf{p})\mathbf{p},$$

for a certain $\alpha \in (0, 1)$. Since $\mathbf{x}^* + \alpha\mathbf{p} \in \mathcal{N}_{\mathbf{x}^*}$, we know that $\nabla^2 f(\mathbf{x}^* + \alpha\mathbf{p})$ is symmetric positive definite, i.e.,

$$\mathbf{p}^T\nabla^2 f(\mathbf{x}^* + \alpha\mathbf{p})\mathbf{p} > 0.$$

Therefore for any $\mathbf{x} = \mathbf{x}^* + \mathbf{p} \in \mathcal{N}_{\mathbf{x}^*}$,

$$f(\mathbf{x}^* + \mathbf{p}) = f(\mathbf{x}^*) + \frac{1}{2}\mathbf{p}^T \nabla^2 f(\mathbf{x}^* + \alpha\mathbf{p})\mathbf{p} > f(\mathbf{x}^*),$$

and $\mathbf{x}^*$ must be a local minimum. $\square$

Applying those results, we can see that $(0, 0)$ is a stationary point for each of the functions $f(\mathbf{x}) = x_1^2 + x_2^2$, $g(\mathbf{x}) = -x_1^2 - x_2^2$, and $h(\mathbf{x}) = x_1^2 - x_2^2$, where it is a minimum of $f(\mathbf{x})$, a maximum of $g(\mathbf{x})$, neither a minimum and a maximum of $h(\mathbf{x})$.

Such results can also be applied to the least squares problems. We solve

$$\min_{\mathbf{x}} f(\mathbf{x}),$$

where the function $f(\mathbf{x})$ is given by

$$f(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2^2 = (\mathbf{b} - A\mathbf{x})^T(\mathbf{b} - A\mathbf{x}) = \mathbf{b}^T\mathbf{b} - 2\mathbf{x}^T A^T\mathbf{b} + \mathbf{x}^T A^T A\mathbf{x}.$$

To find the gradient vector of $f(\mathbf{x})$, we observe that

$$f(\mathbf{x}) = \mathbf{b}^T\mathbf{b} - 2\sum_{i=1}^{n} \left(A^T\mathbf{b}\right)_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \left(A^T A\right)_{i,j} x_i x_j.$$

For any $k = 1, 2, \ldots, n$, the terms related to x_k in $f(\mathbf{x})$ are

$$f(\mathbf{x}) = 2\left(A^T\mathbf{b}\right)_k x_k + \sum_{i=1, i\neq k}^{n} \left(A^T A\right)_{i,k} x_i x_k + \sum_{j=1, j\neq k}^{n} \left(A^T A\right)_{k,j} x_k x_j + \left(A^T A\right)_{k,k} x_k x_k + \ldots$$

Then we can see that

$$\frac{\partial f(\mathbf{x})}{\partial x_k} = -2\left(A^T\mathbf{b}\right)_k + 2\sum_{j=1, j\neq k}^{n} \left(A^T A\right)_{k,j} x_j + 2\left(A^T A\right)_{k,k} x_k = \left(-2A^T\mathbf{b} + 2A^T A\mathbf{x}\right)_k.$$

Therefore

$$\nabla f(\mathbf{x}) = -2A^T\mathbf{b} + 2A^T A\mathbf{x}.$$

Similarly, we can derive that the Hessian of $f(\mathbf{x})$ is

$$\nabla^2 f(\mathbf{x}) = 2A^T A.$$

If the matrix $A \in \mathbb{R}^{m\times n}$ is full rank then we know that the Hessian is symmetric positive definite. If at point $\mathbf{x}^* \in \mathbb{R}^n$, $\nabla f(\mathbf{x}^*) = 0$, then we know from Theorem 8 that $\mathbf{x}^*$ must be a local minimum of $f(\mathbf{x})$. Therefore solution to the least squares problem is equivalent to solve $\nabla f(\mathbf{x}) = 0$, i.e., to solve $A^T A\mathbf{x} = A^T\mathbf{b}$.

4 Algorithms for multidimensional unconstrained optimization

All the optimization algorithms must be iterative. An initial guess $\mathbf{x}_0$ for a local minimum $\mathbf{x}^*$ need to be given, and a sequence of iterates $\mathbf{x}_k$, $k = 0, 1, 2, ...$, will be generated. It

is expected that the sequence $\{\mathbf{x}_k\}$ converge to $\mathbf{x}^*$, as $k \to \infty$. We say the convergence is achieved and terminate the iteration if $\|\nabla f(\mathbf{x}_k)\|_2 \leq \varepsilon$, where ε is a tolerance which can be taken as, e.g., 10^{-6}. Basically, the iteration converges to a stationary point. By enforcing some additional conditions in the algorithm, e.g., by requiring the sequence of function values at the iterates to be decreasing, the iterates converge to a local minimum.

The crucial question in the iteration is how to choose the next iterate $\mathbf{x}_{k+1}$ based on the current iterate $\mathbf{x}_k$. A general approach is to specify a direction $\mathbf{p}_k$, called the *search direction*. Then determine the new iterate $\mathbf{x}_{k+1}$ along the search direction $\mathbf{p}_k$ from $\mathbf{x}_k$, i.e.,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k,$$

where $\alpha_k > 0$, is called the *step size*. α_k represents how far to move along the direction $\mathbf{p}_k$ from $\mathbf{x}_k$. Two questions need be answered: how to choose the search direction $\mathbf{p}_k$ at the current iterate $\mathbf{x}_k$, and how to determine the step size α_k.

Suppose that we are looking for a local minimum of the function. Therefore we certainly expect that the function value at the new iterate be less than the function value at the current iterate, i.e., we expect that

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k).$$

Therefore, we expect that when $\mathbf{x}$ moves from $\mathbf{x}_k$ along the direction $\mathbf{p}_k$, the function value $f(\mathbf{x})$ should be decreasing. This will guide us to choose an appropriate direction $\mathbf{p}_k$. With the descent direction chosen, a line search method will be used to determine how far to move along that direction.

4.1 Steepest descent direction

Given a function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, let us consider how the function value changes at particular iterate $\mathbf{x}_k \in \mathbb{R}^n$ along a given direction $\mathbf{p} \in \mathbb{R}^n$. Without loosing generality, we assume that the length of $\mathbf{p}$ is one, i.e., $\|\mathbf{p}\|_2 = 1$. In this case, along direction $\mathbf{p}$ at $\mathbf{x}_k$, the function $f(\mathbf{x})$ becomes a function of a single variable and can be represented by $f(\mathbf{x}_k + \alpha\mathbf{p})$, where the variable α represents the step size along direction $\mathbf{p}$. Then we can consider the derivative of $f(\mathbf{x}_k + \alpha\mathbf{p})$ with respect to α at $\alpha = 0$, i.e., the rate of change of the function $f(\mathbf{x})$ along direction $\mathbf{p}$ at $\mathbf{x} = \mathbf{x}_k$. We call this derivative the directional derivative of $f(\mathbf{x})$ along the direction $\mathbf{p}$ at $\mathbf{x} = \mathbf{x}_k$, denoted and defined by

$$\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}} = \frac{df(\mathbf{x}_k + \alpha\mathbf{p})}{d\alpha}\Big|_{\alpha=0} = \lim_{\varepsilon \to 0} \frac{f(\mathbf{x}_k + \varepsilon\mathbf{p}) - f(\mathbf{x}_k)}{\varepsilon}.$$

If $\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}}$ is positive, then $f(\mathbf{x})$ increases along the direction $\mathbf{p}$ at $\mathbf{x}_k$; the larger is the value, the faster is the increase. If $\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}}$ is negative, then $f(\mathbf{x})$ decreases along $\mathbf{p}$ at $\mathbf{x}_k$; the larger is the value in negative, the faster is the decrease.

In fact among all the possible directions $\mathbf{p}$ at x_k, there exists a direction along which $\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}}$ achieves its largest value; and along the opposite of which achieves its largest value

in negative. To find such a direction, let us consider the Taylor's expansion from Theorem 6. We have

$$f(\mathbf{x}_k + \varepsilon\mathbf{p}) = f(\mathbf{x}_k) + \varepsilon\nabla f(\mathbf{x}_k)^T\mathbf{p} + \varepsilon^2\frac{1}{2}\mathbf{p}^T\nabla^2 f(\mathbf{x}_k + \alpha\varepsilon\mathbf{p})\mathbf{p},$$

for a certain $\alpha \in (0,\ 1)$. Therefore

$$\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}} = \lim_{\varepsilon\to 0}\frac{f(\mathbf{x}_k + \varepsilon\mathbf{p}) - f(\mathbf{x}_k)}{\varepsilon} = \nabla f(\mathbf{x}_k)^T\mathbf{p}.$$

Assume the direction vector $\mathbf{p}$ has a unit norm, i.e., $\|\mathbf{p}\|_2 = 1$. Then

$$\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}} = \nabla f(\mathbf{x}_k)^T\mathbf{p} = \|\nabla f(\mathbf{x}_k)\|_2\ \|\mathbf{p}\|_2\ \cos\theta = \|\nabla f(\mathbf{x}_k)\|_2\cos\theta,$$

where θ represents the angle between $\mathbf{x}_k$ and $\mathbf{p}$. Then we know that the largest directional derivative $\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}}$ is achieved when $\cos\theta = 1$, i.e., when $\mathbf{p}$ is along the direction $\nabla f(\mathbf{x}_k)$, along which the function value $f(x)$ increases at the fastest speed. Oppositely, when $\cos\theta = -1$, i.e., when $\mathbf{p}$ is opposite to $\nabla f(\mathbf{x}_k)$, $\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}} < 0$ and achieves its largest value in negative. Therefore we call $-\nabla f(\mathbf{x}_k)$ the *steepest descent direction* of $f(\mathbf{x})$ at $\mathbf{x}_k$, along which the function decreases the most rapidly.

In the steepest descent direction method described in the following for solving a unconstrained minimization problem, at each iterate $\mathbf{x}_k$, we choose the search direction as the steepest descent direction, i.e., we take $\mathbf{p}_k = -\nabla f(\mathbf{x}_k)$.

Steepest descent algorithm

Give initial guess x_0
for k = 0, 1, 2, ... until convergence
  $p_k = -\nabla f(x_k)$
  choose appropriate α_k
  $x_{k+1} = x_k + \alpha_k\ p_k$
end

In fact, along any vector $\mathbf{p}_k \in \mathbb{R}^n$ satisfying,

$$\nabla f(\mathbf{x}_k)^T\mathbf{p}_k < 0,\ \text{i.e.,}\ \frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{p}_k} < 0,$$

$f(\mathbf{x})$ decrease along the direction $\mathbf{p}_k$ at $\mathbf{x}_k$. We call such $\mathbf{p}_k$ a descent direction at $\mathbf{x}_k$. Along any descent direction $\mathbf{p}_k$, $f(\mathbf{x})$ decreases at $\mathbf{x}_k$ and we can always find a new iterate $\mathbf{x}_{k+1}$ along the direction $\mathbf{p}_k$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\ \mathbf{p}_k$$

for a certain $\alpha_k > 0$, such that $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

The question then is how to choose an appropriate step size α_k, i.e., how far to move from the current iterate $\mathbf{x}_k$ along the direction $\mathbf{p}_k$ such that $f(\mathbf{x}_{k+1})$ is less than $f(\mathbf{x}_k)$ and at the same time the improvement of the new iterate $\mathbf{x}_{k+1}$ over $\mathbf{x}_k$ is sufficient. We note that the descent direction is only based on the local approximation of the function $f(\mathbf{x})$ at $\mathbf{x}_k$. Moving too far away along the descent direction may not necessarily decrease the function value, while at the same time a too small step may generate only very little improvement. An appropriate step size α_k need be determined.

4.2 Line search procedure

The function $f(\mathbf{x})$ along the direction $\mathbf{p}_k$ at $\mathbf{x}_k$, is just function of a single variable, i.e., just depending on how far to move along the direction $\mathbf{p}_k$ from $\mathbf{x}_k$. Let us denote the step size by α, and define

$$\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k), \quad \alpha > 0,$$

which is just a function of α.

We need to determine α_k and then take $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\ \mathbf{p}_k$. One approach to determine α_k is to solve the one-dimensional minimization problem

$$\alpha_k = \arg\min_{\alpha} \phi(\alpha), \quad \text{i.e.,} \quad \alpha_k = \arg\min_{\alpha} f(\mathbf{x}_k + \alpha\ \mathbf{p}_k).$$

In another word, we determine $\mathbf{x}_{k+1}$ such that $f(\mathbf{x}_{k+1})$ is the minimum of $f(\mathbf{x})$ along the direction $\mathbf{p}_k$ through the current iterate $\mathbf{x}_k$. But such an approach requires solving a one-dimensional minimization problem and can become quite expensive in some applications.

To be more efficient computationally, in stead of solving a one-dimensional minimization problem to determine the step size α_k, we can apply a line search procedure discussed in the following to find a suitable step size α_k.

From this definition of $\phi(\alpha)$, we can see that $\phi(\alpha)$ in fact represents the function $f(\mathbf{x})$ in terms of the step size α along the direction $\mathbf{p}_k$ at $\mathbf{x}_k$. It has the following properties:

$$\phi(0) = f(\mathbf{x}_k), \qquad \phi(\alpha_k) = f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = f(\mathbf{x}_{k+1}),$$

$$\phi'(\alpha) = \nabla f(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k, \quad \text{which implies} \quad \phi'(0) = \nabla f(\mathbf{x}_k)^T \mathbf{p}_k.$$

Since $\mathbf{p}_k$ is a descent direction at $\mathbf{x}_k$, we know that $\phi'(0) < 0$, which means that the function $\phi(\alpha)$ is decreasing at $\alpha = 0$. Therefore it is possible to choose a positive α_k, such that $\phi(\alpha_k) < \phi(0)$, i.e., to find a new iterate $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$, such that $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k)$.

We certainly wish that such a chosen α_k is close to a minimizer of $\phi(\alpha)$. We require that α_k is chosen to satisfy the following two conditions:

- (*Sufficient Decrease Condition*) The chosen $(\alpha_k,\ \phi(\alpha_k))$ point should be below the line $\phi(0) + c_1\phi'(0)\alpha$, where $c_1 \in (0,\ 1)$ is a ceratin given parameter, i.e.,

$$\phi(\alpha_k)\ <\ \phi(0) + c_1\phi'(0)\alpha_k, \quad \text{i.e.,} \quad f(\mathbf{x}_{k+1})\ <\ f(\mathbf{x}_k) + c_1\phi'(0)\alpha_k.$$

This condition guarantees that the decrease of the function value should be proportional to the magnitude of the step size α_k; larger α_k should generate larger reduction in the function value. c_1 is typically a small value, e.g., $c_1 = 10^{-4}$, which means that the line $(\alpha,\ \phi(0) + c_1\phi'(0)\alpha)$ is quite flat typically, i.e., the enforcement of such condition is quite loose.

- (*Curvature Condition*) For a ceratin given parameter $c_2 \in (0,\ 1)$,

$$|\phi'(\alpha_k)| \ \leq \ c_2|\phi'(0)|.$$

Such a condition is motivated by requiring that α_k be close to a minimizer of the function $\phi(\alpha)$, where $\phi'(\alpha)$.

We notice that typically the enforcement of such two conditions are quite generous by choosing appropriate c_1 and c_2, e.g., $c_1 = 10^{-4}$ and $c_2 = 0.9$ as mentioned above. We call the above two conditions in the *line search procedure* the *Wolfe conditions*.

The following theorem tells the existence of α_k satisfying the Wolfe conditions.

Theorem 9 *Let that $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable. Let $\mathbf{p}_k$ be a descent direction at $\mathbf{x}_k$. Assume that $f(\mathbf{x}_k + \alpha\mathbf{p}_k)$ is bounded below for all $\alpha > 0$. Then if $0 < c_1 < c_2 < 1$, there always exist intervals of step size satisfying the Wolfe conditions.*

The following theorem tells that the sequence of iterates $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$, with the step size α_k determined by the line search procedure enforcing the Wolfe conditions, always converges to a local minimum of $f(\mathbf{x})$, as long as the search directions are descent directions at each iterate.

Theorem 10 *Let $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ be a smooth function. The sequence*

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k, \quad k = 0, 1, 2, \ldots$$

always converges to a local minimum of $f(\mathbf{x})$, if at each iterate $\mathbf{x}_k$, the search direction $\mathbf{p}_k$ is a descent direction and α_k satisfies the Wolfe conditions.

The line search algorithm, which finds such a step size α_k satisfying the Wolfe conditions, includes two stages. The first stage begins with a trial estimate α_1, and keeps increasing it until it finds either a step length which satisfies the Wolfe conditions, or an interval that brackets the desired step lengths. In the later case, a second stage, **lszoom**$(\alpha_{lo}, \alpha_{hi})$, implements the *backtracking* until an acceptable step length is found.

Line Search Algorithm (part I)

Set $\alpha_0 = 0$, $\alpha_1 = 1$**, and a maximum number of line search steps,** $nsteps = 10$**,**
for $i = 1$ **to** $nsteps$
  if $(\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0))$ **or** $(\phi(\alpha_i) > \phi(\alpha_{i-1}))$
    $\alpha_* =$ **lszoom**(α_{i-1}, α_i)**;** **return;**

if $|\phi'(\alpha_i)| \le c_2|\phi'(0)|$
$\alpha_* = \alpha_i$; **return;**
if $\phi'(\alpha_i) \ge 0$
$\alpha_* =$ **lszoom**(α_i, α_{i-1}); **return;**
$\alpha_{i+1} = 2\alpha_i$;
end
Error('step size α_k **not found within 10 iterations')**

Line Search Algorithm (part II, $\alpha_* =$ **lszoom**$(\alpha_{lo}, \alpha_{hi})$ **)**

Choose a maximum number of backtracking steps, $nsteps = 50$**,**
for $i = 1$ **to** $nsteps$
$\alpha_m = (\alpha_{lo} + \alpha_{hi})/2$;
if $(\phi(\alpha_m) > \phi(0) + c_1\alpha_m\phi'(0))$ **or** $(\phi(\alpha_m) > \phi(\alpha_{lo}))$
$\alpha_{hi} = \alpha_m$;
else
if $|\phi'(\alpha_m)| \le c_2|\phi'(0)|$
$\alpha_* = \alpha_m$; **return;**
if $\phi'(\alpha_m)(\alpha_{hi} - \alpha_{lo}) \ge 0$
$\alpha_{hi} = \alpha_{lo}$;
end
$\alpha_{lo} = \alpha_m$;
end
end
Error('step length α_k **not found within 50 iterations')**

From the line search algorithm, we can see that the following are always true for α_{lo}, α_{hi} in the second stage:

1. It is not necessary that $\alpha_{lo} < \alpha_{hi}$. But $\phi(\alpha_{lo}) \le \phi(\alpha_{hi})$ always hold. In fact, α_{lo} is the one having the smallest function value among all step lengths generated so far satisfying the sufficient decrease condition.
2. The interval bounded by α_{lo} and α_{hi} always bracket the acceptable step sizes.

4.3 Newton's methods

In the steepest descent method, at each iterate $\mathbf{x}_k$, the search direction $\mathbf{p}_k$ is chosen as the steepest descent direction

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k).$$

Then a line search method is used to determine the step size α_k and the new iterate is determined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k.$$

However the steepest descent direction $-\nabla f(\mathbf{x}_k)$ is derived based on a linear approximation of the function $f(\mathbf{x})$ around the iterate $\mathbf{x}_k$. It is not necessarily always a good choice. For example consider the minimization of the function $f(\mathbf{x}) = x_1^2 + 100x_2^2$. We know that the minimum is achieved at the origin. However when a steepest descent direction method starts from, e.g., $\mathbf{x}_0 = (1, 1)$, the steepest descent direction $\mathbf{p}_0 = -\nabla f(\mathbf{x}_0)$ points almost parallel to the y-axis. Overall all, oscillatory behavior of the convergence will often be observed for the steepest descent method.

In this section, we consider another choice of the descent direction in the minimization algorithms, the Newton's direction.

Let us consider the quadratic approximation of the function $f(\mathbf{x})$ in a neighborhood of $\mathbf{x}_k$. From Theorem 6, we know that $f(\mathbf{x})$ can be approximated by

$$f(\mathbf{x}) \approx M(\mathbf{x}) := f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^T \nabla f(\mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \nabla^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k),$$

in neighborhood of $\mathbf{x}_k$. The stationary point of the quadratic function $M(\mathbf{x})$ is found by solving

$$\nabla M(\mathbf{x}) = 0,$$

which is

$$\nabla^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) = -\nabla f(\mathbf{x}_k),$$

and gives the solution

$$\mathbf{x} = \mathbf{x}_k - \nabla^2 f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k).$$

The point $\mathbf{x}$ could be a minimum, a maximum, or a saddle point of $M(\mathbf{x})$ depending on the property of the Hessian $\nabla^2 f(\mathbf{x}_k)$. If $\nabla^2 f(\mathbf{x}_k)$ is positive definite, then $M(\mathbf{x})$ is concave upward and $\mathbf{x}$ is a minimum of $M(\mathbf{x})$. In this case, it is then reasonable to take $\mathbf{x}$ as the new iterate, i.e.,

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

and we call

$$\mathbf{p}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k)$$

the Newton's direction. More generally, if $\nabla^2 f(\mathbf{x}_k)$ is positive definite, then $\mathbf{p}_k$ is a descent direction and a line search can be performed along the Newton's direction to determine the step size and the new iterate. We have the following Newton's algorithm.

Newton's algorithm for minimizing $f(x)$

Give initial guess x_0
for $k = 0, 1, 2, \ldots$ **until convergence**
  Solve $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$ **for** p_k
  Choose α_k **by line search**

$x_{k+1} = x_k + \alpha_k \; p_k$

end

As Newton's method for one-dimensional problems, when it converges, the convergence is quadratic when the iterate is close to the true solution. It is a lot faster than the steepest descent method. For example, to minimize the function, $f(\mathbf{x}) = x_1^2 + 100x_2^2$, it only takes one Newton iteration to find the minimum.

However, in general, the Newton's method is not guaranteed to converge. It may happen at a certain iterate $\mathbf{x}_k$ that the Hessian $\nabla^2 f(\mathbf{x}_k)$ is not positive definite. Then the Newton's direction $\mathbf{p}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k)$ is not necessarily a descent direction, in which case the line search algorithm may fail.

To guarantee the convergence of Newton's method to a local minimum of a function, the Newton's search direction $\mathbf{p}_k$ needs to be modified such that it is always a descent direction and ideally close to the original Newton's direction when the Newton's direction is a descent direction. We call such methods with modified Newton's direction the quasi-Newton method. There are different ways to modify $\mathbf{p}_k$. Typically, at each current iterate $\mathbf{x}_k$, the quasi-Newton direction is determined in the form of

$$\mathbf{p}_k = -B_k^{-1} \nabla f(\mathbf{x}_k).$$

Here we require that B_k is always positive definite such that $\mathbf{p}_k$ is always a descent direction. We certainly hope that B_k approximate the Hessian matrix $\nabla^2 f(\mathbf{x}_k)$ reasonably well, in particular when $\nabla^2 f(\mathbf{x}_k)$ is in fact positive definite.

In the following, we discuss two quasi-Newton approaches.

4.4 Hessian Modification

The most straight forward approach is to modify the Hessian matrix $\nabla^2 f(\mathbf{x}_k)$ to generate a sufficiently positive definite matrix B_k. Let us denote the smallest eigenvalue of $\nabla^2 f(\mathbf{x}_k)$ by λ_{min}. Let δ be a positive value, e.g., $\delta = 10^{-6}$. If $\lambda_{min} \geq \delta$, then all the eigenvalues of $\nabla^2 f(\mathbf{x}_k)$ are positive and $\nabla^2 f(\mathbf{x}_k)$ is symmetric positive definite. In this case, we just take $B_k = \nabla^2 f(\mathbf{x}_k)$. If $\lambda_{min} < \delta$, then $\nabla^2 f(\mathbf{x}_k)$ is not sufficiently positive definite (can even be indefinite). In this case, we take

$$B_k = \nabla^2 f(\mathbf{x}_k) + (\delta - \lambda_{min}) I,$$

i.e., B_k is obtained by shifting the eigenvalues of $\nabla^2 f(\mathbf{x}_k)$ to the right for a distance of $(\delta - \lambda_{min})$. Then the smallest eigenvalue of B_k is δ and B_k is sufficiently positive definite.

We have the following algorithm with this Hessian modification.

Quasi-Newton algorithm with Hessian modification

Give initial guess x_0**, choose** $\delta > 0$
for $k = 0, 1, 2, \ldots$ **until convergence**

if $\lambda_{min}(\nabla^2 f(x_k)) \geq \delta$

$B_k = \nabla^2 f(x_k)$

else

$B_k = \nabla^2 f(x_k) + (\delta - \lambda_{min})I$

end

solve $B_k p_k = -\nabla f(x_k)$ **for** p_k

choose α_k **by line search**

$x_{k+1} = x_k + \alpha_k \; p_k$

end

The Hessian modification is simple, but it is also very expensive due to solving the eigenvalue problems. In the following, we discuss another approach which is much cheaper.

4.5 BFGS method

Let us now consider that in a Newton's iteration, after obtaining $\mathbf{x}_{k+1}$, we need to determine a search direction $\mathbf{p}_{k+1}$. The Newton's direction is $-\nabla^2 f(\mathbf{x}_{k+1})^{-1}\nabla f(\mathbf{x}_{k+1})$. Here let us consider an approximation of the Hessian $\nabla^2 f(\mathbf{x}_{k+1})$.

The Taylor expansion of $\nabla f(\mathbf{x}_k)$ at the point $\mathbf{x}_{k+1}$ can be written as

$$\nabla f(\mathbf{x}_k) \approx \nabla f(\mathbf{x}_{k+1}) + \nabla^2 f(\mathbf{x}_{k+1})(\mathbf{x}_k - \mathbf{x}_{k+1}),$$

i.e.,

$$\nabla^2 f(\mathbf{x}_{k+1})(\mathbf{x}_{k+1} - \mathbf{x}_k) \;\approx\; \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k).$$

Denote

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \quad \mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k).$$

Let B_{k+1} be an approximation of the Hessian $\nabla^2 f(\mathbf{x}_{k+1})$. Then it is reasonable to require that

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k.$$

If we can find such a matrix B_{k+1}, which is also symmetric positive definite, then we can use B_{k+1} as an approximation of $\nabla^2 f(\mathbf{x}_{k+1})$ and take take

$$\mathbf{p}_{k+1} = -B_{k+1}^{-1} \nabla f(\mathbf{x}_{k+1}).$$

as the modified Newton search direction at iterate $\mathbf{x}_{k+1}$. $\mathbf{p}_{k+1}$ is then a descent direction at $\mathbf{x}_{k+1}$ and is also close to the Newton's direction $-\nabla^2 f(\mathbf{x}_{k+1})^{-1} \nabla f(\mathbf{x}_{k+1})$.

Before we discuss an algorithm to find such a matrix B_{k+1}, let us first look at if a necessary condition for B_{k+1} being symmetric positive definite is true. In order that B_{k+1} satisfies $B_{k+1}\mathbf{s}_k = \mathbf{y}_k$ and is symmetric positive definite, it should hold that

$$\mathbf{y}_k^T \mathbf{s}_k > 0,$$

i.e.,

$$\left(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\right)^T (\mathbf{x}_{k+1} - \mathbf{x}_k) \ > \ 0.$$

This condition can be proved as follows. From the Wolfe conditions, we know that

$$|\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k| \leq c_2 |\nabla f(\mathbf{x}_k)^T \mathbf{p}_k|.$$

Since $\mathbf{s}_k = \alpha_k \mathbf{p}_k$, and it is a descent direction, we have

$$|\nabla f(\mathbf{x}_{k+1})^T \mathbf{s}_k| \leq c_2 |\nabla f(\mathbf{x}_k)^T \mathbf{s}_k| = -c_2 \nabla f(\mathbf{x}_k)^T \mathbf{s}_k,$$

Therefore

$$\mathbf{y}_k^T \mathbf{s}_k = \left(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\right)^T \mathbf{s}_k \geq (c_2 - 1) \nabla f(\mathbf{x}_k)^T \mathbf{s}_k > 0,$$

where the parameter $c_2 < 1$. Therefore the necessary condition for B_{k+1} being symmetric positive definite indeed holds. But it does not tells how to determine B_{k+1}.

In the following, we explain how to determine a symmetric positive definite matrix B_{k+1} to satisfy the secant equation

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k,$$

with given $\mathbf{s}_k$ and $\mathbf{y}_k$.

Given $\mathbf{s}_k$ and $\mathbf{y}_k$ in $\mathbb{R}^n$, to determine a $n \times n$ symmetric positive definite matrix B_{k+1} to satisfy

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k,$$

we need to determine the values of the $n(n+1)/2$ entries of B_{k+1} (since it is symmetric). Now let us compute how many constraints we have to determine B_{k+1}. The secant equation gives us n constarins, the positive definiteness gives us an additional n constraints. In total we have $2n$ constraints to determine $n(n+1)/2$ unknowns, which means the choice of B_{k+1} is not unique.

To determine B_{k+1} uniquely, we impose additional conditions such that, among all symmetric positive definite matrices satisfying the secant equation, B_{k+1} is the closest one to the current matrix B_k, i.e., we solve

$$\min_{B\in\mathbb{R}^{n\times n}} \ \|B - B_k\|$$

$$\text{subject to} \quad B = B^T > 0 \ \text{ and } \ B\mathbf{s}_k = \mathbf{y}_k,$$

for B_{k+1}.

From the observation above, to compute $\mathbf{p}_{k+1} = -B_{k+1}^{-1}\nabla f(\mathbf{x}_{k+1})$, what we really need is to multiply B_{k+1}^{-1} with a vector. Therefore it is better to determine B_{k+1}^{-1} directly through such an approach than determining B_{k+1} and solving a system of linear equation. Due to such consideration, let us find H_{k+1}, such that

$$H_{k+1}\mathbf{y}_k = \mathbf{s}_k,$$

and H_{k+1} is symmetric positive definite. Here H_{k+1} plays the role of B_{k+1}^{-1} essentially. Since such conditions are not sufficient to determine H_{k+1} uniquely, as for the case of finding B_{k+1}, we need to solve

$$\min_{H\in\mathbb{R}^{n\times n}} \ \|H - H_k\|$$

$$\text{subject to} \quad H = H^T > 0 \ \text{ and } \ H\mathbf{y}_k = \mathbf{s}_k.$$

The initial H_0 at the initial step can be taken as the identity matrix.

By choosing different matrix norms, different solutions will be obtained. Among the most popular ones is the following BFGS formula

BFGS algorithm

Given initial guess x_0. **Take** $H_0 = I$ **(the identity matrix).**
for $k = 0, 1, 2, \ldots$ **until convergence**

choose $p_k = -H_k\nabla f(x_k)$

choose α_k **using line search**

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

$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k), \quad \rho_k = \dfrac{1}{y_k^T s_k}$

$H_{k+1} = (I - \rho_k s_k y_k^T)H_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$

end

---

5 Solving Nonlinear Equations

We have discussed solving nonlinear equations of a single variable, i.e., for a given function $f(x) : \mathbb{R} \to \mathbb{R}$, we considered solving the equation

$$f(x) = 0.$$

The interval bisection method and the Newton's method have been discussed. For example, the iterates generated by the Newton's method are given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots$$

Also to make the convergence of Newton's method robust, an interval bisection strategy can be added into the Newton's algorithm.

Now let us consider solving a system of nonlinear equations

$$F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ has n components and each component $f_j(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is a function of $\mathbf{x}$, which contains n variables. Such solutions are also called the roots of $F(\mathbf{x})$.

Solutions of nonlinear equations, especially systems of nonlinear equations, are complicated. In general a system of nonlinear equations may have no solution, a unique solution, or any number of solutions. Here we discuss numerical approaches to solve such nonlinear equations, if such a solution exists. The basic approach is still the Newton's method. But the interval bisection approach to make it globally convergent is no longer applicable in the multidimensional case.

In the following, we first derive the Newton's iterate for solving systems of nonlinear equations and then we discuss its connection with solving the unconstrained minimization problems that we have discussed earlier.

Given a function $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$, its Jacobian $J(x)$ is defined by

$$J(x) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ & & \cdots & \\ \frac{\partial f_n(\mathbf{x})}{\partial x_1} & \frac{\partial f_n(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x})}{\partial x_n} \end{bmatrix} = [\ \nabla f_1(\mathbf{x})\ \nabla f_1(\mathbf{x})\ \ldots\ \nabla f_n(\mathbf{x})\ ]^T.$$

which essentially represents the first derivative information of the function $F(\mathbf{x})$.

The following theorem on Taylor's expansion is essentially the extension of the Taylor's expansion

$$f(x+h) \ = f(x) + \int_0^1 \ f'(x+th)h \ dt, \text{ i.e., } f(x+h) - f(x) \ = \ \int_0^1 \ f'(x+th)h \ dt,$$

to multi-dimensional case.

Theorem 11 *Let $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable. For any $\mathbf{x}, \mathbf{x}+\mathbf{p} \in \mathbb{R}^n$,*

$$F(\mathbf{x}+\mathbf{p}) = F(\mathbf{x}) + \int_0^1 J(\mathbf{x}+t\mathbf{p})\mathbf{p}dt,$$

where $J(\mathbf{x})$ is the Jacobian of $F(\mathbf{x})$ and the integration is taken for each component of the vector.

Given a current iterate $\mathbf{x}_k$ in an iterative method for finding the root of $F(\mathbf{x})$, from Theorem 11, we have

$$F(\mathbf{x}) = F(\mathbf{x}_k) + \int_0^1 J(\mathbf{x}_k + t(\mathbf{x}-\mathbf{x}_k))(\mathbf{x}-\mathbf{x}_k)dt.$$

Taking an approximation of the integral, the function $F(\mathbf{x})$ can be approximated in a neighborhood of $\mathbf{x}_k$ by a linear function

$$M(\mathbf{x}) = F(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k).$$

Another way to derive the above approximation of $F(\mathbf{x})$ is that each component $f_i(\mathbf{x})$ of $F(\mathbf{x})$ can be approximated by $f_i(\mathbf{x}_k) + \nabla f_i(\mathbf{x}_k)^T(\mathbf{x}-\mathbf{x}_k)$, which all together form the approximation $F(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)$.

We can then take the root of the linear approximation $M(\mathbf{x})$ as the new iterate in the algorithm to find a root of $F(\mathbf{x})$, i.e., we take

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J(\mathbf{x}_k)^{-1}F(\mathbf{x}_k).$$

We call such iteration the Newton's algorithm for solving the system of nonlinear equations $F(\mathbf{x}) = \mathbf{0}$.

Newton's algorithm for solving $F(x) = 0$

Given initial guess x_0
for $k = 0, 1, 2, \ldots$ **until convergence**

  solve $J(x_k)p_k = -F(x_k)$ **for** p_k

  $x_{k+1} = x_k + p_k$

end

If the Jacobian matrix is non-singular in a neighborhood of a root of $F(\mathbf{x})$ and the initial guess $\mathbf{x}_0$ is close enough to the root, then the Newton's method converge to the root quadratically. However, application of the above Newton's method is not robust, e.g., when the Jacobian matrix $J(\mathbf{x}_k)$ is close to be singular, or when the initial guess $\mathbf{x}_0$ is far away from a root.

In order to make the Newton's method for solving nonlinear equations globally convergent, we need to discuss a connection between the solution of nonlinear equations and the unconstrained minimization problems. Such connections make it straightforward to apply the global convergent strategy in the minimization problems to solving nonlinear equations.

Since a root of $F(\mathbf{x})$ is a minimizer of the function

$$\frac{1}{2}\|F(\mathbf{x})\|_2^2,$$

which is

$$\frac{1}{2}F(\mathbf{x})^T F(\mathbf{x}),$$

we can consider solution strategies of the unconstrained minimization problem

$$\min_{\mathbf{x}} \ \frac{1}{2}\|F(\mathbf{x})\|_2^2.$$

We can see that any solution to $F(\mathbf{x}) = 0$ is a minimizer of the $\|F(\mathbf{x})\|_2^2$, even though some local minimizers of $\|F(\mathbf{x})\|_2^2$ is not necessarily a solution to $F(\mathbf{x}) = 0$. Therefore, we still wish to keep using the Newton's iterate for solving nonlinear equations as much as possible, but we can of course use the objective function $\|F(\mathbf{x})\|_2^2$ in a line search procedure to make sure that the value $\|F(\mathbf{x})\|_2^2$ decrease at each new iterate. Therefore we have the following Newton's algorithm using line search.

Newton's algorithm with line search for solving $F(x) = 0$

Given initial guess x_0
for $k = 0, 1, 2, \ldots$ **until convergence**

solve $J(x_k)p_k = -F(x_k)$ **for** p_k

determine α_k **by line search with objective function** $\frac{1}{2}\|F(x)\|_2^2$

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

end

The above algorithm is essentially an algorithm to find local minimum of the function $\frac{1}{2}\|F(\mathbf{x})\|_2^2$. Then there are two questions. First, is the search direction $\mathbf{p}_k =$

$-J(\mathbf{x}_k)^{-1}F(\mathbf{x}_k)$ always a descent direction of the function $\frac{1}{2}\|F(\mathbf{x})\|_2^2$ at $\mathbf{x}_k$? Second, when the algorithm converges to a local minimum of $\frac{1}{2}\|F(\mathbf{x})\|_2^2$, is that local minimum a root of $F(\mathbf{x})$?

To answer the first question, we observe that the gradient vector of the function $f(\mathbf{x}) = \frac{1}{2}F(\mathbf{x})^T F(\mathbf{x})$ equals

$$\begin{aligned}\nabla f(\mathbf{x}) &= \nabla\left(\frac{1}{2}F(\mathbf{x})^T F(\mathbf{x})\right) = \frac{1}{2}\nabla\left(\sum_{i=1}^{n} f_i(\mathbf{x})^2\right) = \frac{1}{2}\sum_{i=1}^{n}\nabla\left(f_i(\mathbf{x})^2\right) \\ &= \sum_{i=1}^{n} f_i(\mathbf{x})\nabla f_i(\mathbf{x}) = J(\mathbf{x})^T F(\mathbf{x}).\end{aligned}$$

At each current iterate $\mathbf{x}_k$, as long as $J(\mathbf{x}_k)$ is nonsingular and $F(\mathbf{x}_k) \neq 0$, we have

$$\nabla f(\mathbf{x}_k)^T \mathbf{p}_k = -F(\mathbf{x}_k)^T J(\mathbf{x}_k) J(\mathbf{x}_k)^{-1} F(\mathbf{x}_k) = -F(\mathbf{x}_k)^T F(\mathbf{x}_k) < 0.$$

Therefore as long as all $J(\mathbf{x}_k)$ are nonsingular, the Newton's direction $\mathbf{p}_k = -J(\mathbf{x}_k)^{-1}F(\mathbf{x}_k)$ is always a descent direction of the function $\frac{1}{2}\|F(\mathbf{x})\|_2^2$ at $\mathbf{x}_k$ and the iterate sequence $\{\mathbf{x}_k\}$ converges to $\mathbf{x}^*$ where $\nabla f(\mathbf{x}^*) = 0$.

To answer the second question, since at the converged point $\mathbf{x}^*$, $\nabla f(\mathbf{x}^*) = 0$, and

$$\nabla f(\mathbf{x}^*) = J(\mathbf{x}^*)^T F(\mathbf{x}^*),$$

we know that $F(\mathbf{x}^*) = 0$ as long as $J(\mathbf{x}^*)$ is nonsingular.

If at an iteration step, $J(\mathbf{x}_k)$ becomes singular, then some procedures similar to the Hessian modifications for the unconstrained optimization problem can be applied here, which will not be addressed here.