OR4030 OPTIMIZATION Chapter 3
One-Dimensional Unconstrained Optimization — Line Search Methods
3.1 Introduction

3,1.1 Problem Description

Problem:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & x \in S = [x_s, x_t] \subset \Re, \end{aligned}$$

where

1. $f : \Re \mapsto \Re$;
2. $[x_s, x_t]$ is a given interval of uncertainty;

3.1.2 A Trivial Example

Problem:

$$\begin{aligned} \min \quad & f(x) = 12x^6 + 3x^4 - 12x + 7, \\ \text{s.t.} \quad & 0 \le x \le 1. \end{aligned}$$

We know that

$$f'(x) = 72x^5 + 12x^3 - 12, \quad \text{and} \quad f''(x) = 360x^4 + 36x^2.$$

If we can solve the equation

$$f'(x) = 72x^5 + 12x^3 - 12 = 0, \tag{1}$$

and if a root x^* of equation (1) is in the interval of $[0, 1]$, then it is a stationary point. Furthermore, if $x^* \neq 0$, then $f''(x^*) > 0$, and it must be a minimizer. In fact the optimal solution is

$$x^* = 0.65435605252093 \quad \text{with} \quad f(x^*) = 0.63977916586877.$$

However, as (1) is an order 5 equation, it is not easy to obtain the solution.

3.1.3 Remarks

- When analytical solution is not achievable, numerical methods are essential to the success of finding an optimum solution.
- The process of determining the minimum point on a given line (i.e. with only one variable) is called *line search* or *one dimensional search*. The line search techniques are procedures for solving one-dimensional minimization problems.

3.2 One-Dimensional Search Methods that Use Function Values Only

3.2.1 Unimodal Functions

In this section we consider minimization for unimodal functions.

Definition Let x^* be the minimum point of f over an interval $[x_s, x_t]$. f is called a *unimodal* function on $[x_s, x_t]$ if for any x_1, x_2, x_3 and x_4 in $[x_s, x_t]$ with the arrangement:

$$x_s < x_1 < x_2 < x^* < x_3 < x_4 < x_t$$

then their function values must have the the following relationship:

$$f(x_s) > f(x_1) > f(x_2) > f(x^*) < f(x_3) < f(x_4) < f(x_t).$$

Or simply speaking, to the right of x^*, f is an increasing one, whereas to the left of x^*, f is a decreasing one.

Consider a unimodal function defined on an interval $[x_s, x_t]$. Assume that two function evaluations are carried out at x_1 and x_2 with $x_s < x_1 < x_2 < x_t$.

- If $f(x_1) < f(x_2)$, then $[x_2, x_t]$ can be discarded, because for each $x \in [x_2, x_t]$, its function value $f(x) \geq f(x_2)$. Hence $[x_2, x_t]$ cannot contain the minimum point. By the same reasoning,
- If $f(x_1) > f(x_2)$, then $[x_s, x_1]$ can be discarded.
- If $f(x_1) = f(x_2)$, then both $[x_s, x_1]$ and $[x_2, x_t]$ can be discarded.

The above conclusions mean that by comparing two function values in $[x_s, x_t]$, we can reduce the search interval by discarding a part of the interval. Based on this property, we may design a class of methods for locating the minimum point for a unimodal function.

Output of this class of line search methods

The output is the final *interval* of uncertainty, not the exact minimum point. But we often approximately take the middle point of the final interval as the minimum point.

To design algorithms for minimizing unimodal functions, we should consider to calculate function values *at what points*, so that we can let the final interval of uncertainty meet the required accuracy while the number of function evaluations can be as small as possible.

3.2.2 Exhaustive Search Method

- Evaluating the objective function at a predetermined number (N) of equally spaced points in the interval (x_s, x_t),

$$x_s < x_1 < \cdots < x_N < x_t.$$

- If the minimum value among the N function values is x_K, then the final interval of uncertainty is $[x_{K-1}, x_{K+1}]$ with length of

$$L_N = x_{K+1} - x_{K-1} = \left(\frac{2}{N+1}\right) L_0,$$

where $L_0 = x_t - x_s$.

- If we ask the length of the final interval of uncertainty to be bounded by $\varepsilon > 0$, then we should let

$$(\frac{2}{N+1})L_0 < \varepsilon \Longrightarrow N+1 > \frac{2L_0}{\varepsilon} \Longrightarrow N > \frac{2L_0}{\varepsilon} - 1.$$

Advantage

- The algorithm is simple.

Disadvantages

- N is predetermined. That is, after computing the function values for N times, the procedure cannot continue by one extra function value calculation.
- This is a simultaneous search method (i.e., no decision can be made by only one of these N function evaluations) and is relatively inefficient.

Algorithm (Exhaustive Search Method)

Step 1. Input and Initialization:

(a) Input $[x_s, x_t]$ = the initial interval of uncertainty.

(b) Input N (≥ 2) = the total number of function evaluations to be conducted.

(c) Let $L_0 := x_t - x_s$.

Step 2. Main procedure:

(a) Generate N equally spaced points in the interval (x_s, x_t), i.e.,

$$x_k := x_s + \left(\frac{k}{N+1}\right) L_0, \text{ for } k = 1, 2, \ldots, N.$$

(b) Calculate all $f(x_k)$ and find J such that $f(x_J) = \min_k f(x_k)$.

Step 3. Output $[x_{J-1}, x_{J+1}]$ = the final interval of uncertainty.

Golden Section Search Method

Idea

- We know that the positive root to the equation

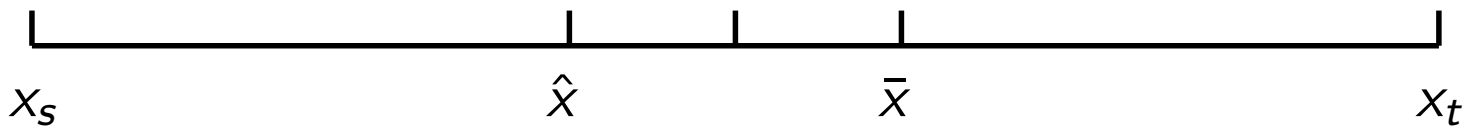
$$\tau^2 + \tau - 1 = 0$$

is $\tau = \frac{-1+\sqrt{5}}{2} \approx 0.618$. This τ is called the golden ratio.

- Let the interval of uncertainty be $[x_s, x_t]$, in which we choose the first test point

$$\bar{x} = x_s + \tau L_0, \tag{2}$$

where L_0 is the length of the interval: $L_0 = x_t - x_s$. We next choose the second test point $\hat{x}$ as the point symmetric to $\bar{x}$ with respect to the middle point of the interval $[x_s, x_t]$. How to express the location of $\hat{x}$ analytically?



Due to symmetry,

$$|x_s\hat{x}| = |\bar{x}x_t| = (1-\tau)L_0.$$

So,

$$\hat{x} = x_s + |x_s\hat{x}| = x_s + (1-\tau)L_0. \tag{3}$$

- We calculate $f(\bar{x})$ and $f(\hat{x})$, and compare their values.
  Case A If $f(\bar{x}) \geq f(\hat{x})$, we discard $[\bar{x}, x_t]$ and let the remaining interval of uncertainty be

$$[x_s^1, x_t^1] = [x_s, \bar{x}]; \tag{4}$$

  Case B otherwise, $f(\bar{x}) < f(\hat{x})$, we discard $[x_s, \hat{x}]$ and let

$$[x_s^1, x_t^1] = [\hat{x}, x_t]. \tag{5}$$

- The first iteration is finished and we obtain the second interval of uncertainty $[x_s^1, x_t^1]$. Let its length be L_1. L_1=? Note that in either of the above two cases, the removed length is $(1-\tau)L_0$. Hence

$$L_1 = L_0 - (1-\tau)L_0 = \tau L_0. \tag{6}$$

That is, after this iteration, the length of the shortened interval is reduced to the τ times of the original length.

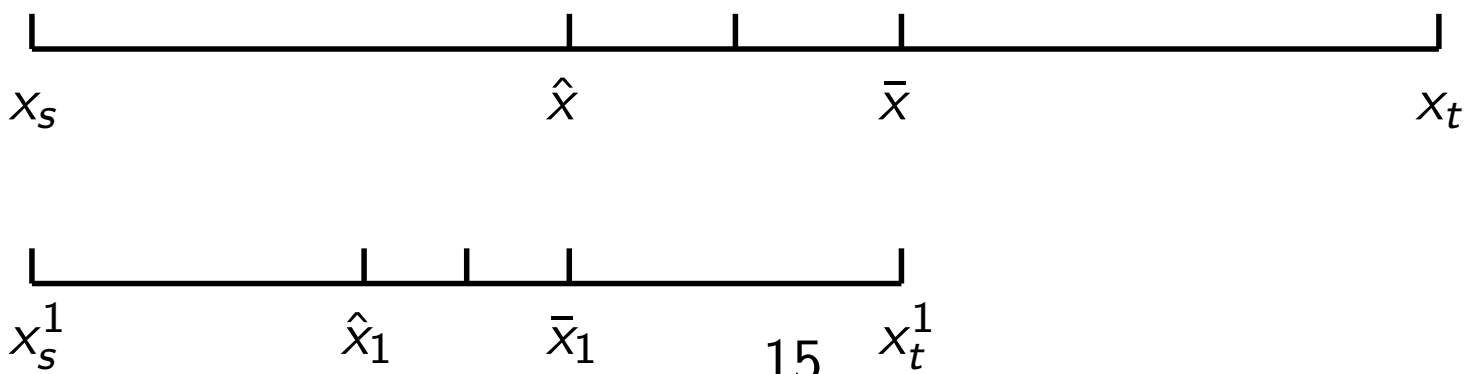
- Consider the second iteration. Suppose it is the Case A:

$$[x_s^1, x_t^1] = [x_s, \bar{x}],$$

and we need to arrange two test points, $\bar{x}_1$ and $\hat{x}_1$, in $[x_s^1, x_t^1]$ by the same method, that is, we let

$$\bar{x}_1 = x_s^1 + \tau L_1$$

and $\hat{x}_1$ be the point symmetric to $\bar{x}_1$ with respect to the mid-point of the interval $[x_s^1, x_t^1]$.



- By formula (2), we see that

$$\begin{aligned} \bar{x}_1 &= x_s^1 + \tau L_1 \\ &= x_s + \tau^2 L_0 \\ &= x_s + (1-\tau) L_0 \quad (\text{because } \tau^2 = 1 - \tau) \\ &= \hat{x}. \quad (\text{see formula (3) }) \end{aligned}$$

Hence, $\bar{x}_1$ and $\hat{x}$ coincide so that $f(\bar{x}_1) = f(\hat{x})$, which has already been calculated in the last iteration. On the other hand, by formula (3),

$$\hat{x}_1 = x_s^1 + (1-\tau) L_1,$$

which is a new point. We calculate $f(\hat{x}_1)$, then by comparing the values $f(\bar{x}_1)$ and $f(\hat{x}_1)$, we can reduce the interval of uncertainty to a new one $[x_s^2, x_t^2]$ (like the formulas (4) and (5)).

We have completed the second iteration. Like (6), the length L_2 of the interval $[x_s^2, x_t^2]$ is

$$L_2 = \tau L_1. \tag{7}$$

Note that the **important advantage** of the Golden Section method is that starting with the second iteration, between the two test points, one is used in the previous iteration so that its function value is readily available. Hence in order to compare the two values, we only need to calculate ONE, not two, function values. This idea helps the method enhance efficiency.

An equivalent way to express this advantage is that if we set point $\bar{x}$ by formula (2) as the initial test point before all iterations, then in each iteration (including the first one), the Golden Section method need to generate and evaluate only one test point.

- The above analysis shows that in the method the lengths of intervals will decrease with a constant rate:

$$L_k = \tau L_{k-1}, \quad k = 1, 2, \ldots$$

So, if we repeat the procedure for N times, then the length of final interval of uncertainty is

$$L_N = \tau L_{N-1} = \tau^2 L_{N-2} = \cdots = \tau^N L_0.$$

Using this formula, we may determine the number N of iterations in order to let the length of final interval be within a given accuracy ε:

$$L_N \leq \varepsilon \Longrightarrow \tau^N \leq \frac{\varepsilon}{L_0} \Longrightarrow N \geq \log_\tau(\frac{\varepsilon}{L_0}).$$

And we know that to conduct N iterations, we need to compute function values for $N + 1$ times.

Note that another advantage of the Golden Section method, comparing to the exhaustive method, is that even if we have finished N iterations, if we wish to make more iterations to raise accuracy further, we can repeat the same procedure for any number of times.

Below we state the algorithm. In the algorithm, Step 1 and Step 2 are initial steps in which the first test point, denoted by $\bar{x}^1$, is given by formula (2). Step 3 is the main step which shall repeat for N times, giving N iterations. In the k-th iteration, $\bar{x}^k$ is the test point which we already know in the last iteration, and x_{k+1} is the new test point which is symmetric to $\bar{x}^k$ in the current interval of uncertainty.

Algorithm (Golden Section Method)

Step 1. Input and Initialization:

(a) Input $[x_s^0, x_t^0]$ = the initial interval of uncertainty.

(b) Input N (≥ 2) = the total number of iterations to be conducted.

(c) Let $L_0 := x_t^0 - x_s^0$.

(d) Let $\tau := \dfrac{\sqrt{5}-1}{2} \approx 0.618$ = the golden ratio.

Step 2. Place the first point:

(a) Let $\bar{x}^1 := x_s^0 + \tau L_0$.

(b) Let $x_1 := \bar{x}^1$.

(c) calculate $f(x_1)$.

Step 3. For $k = 1, 2, 3, \ldots, N$, do the following:

(a) Place a point x_{k+1} symmetrically in the interval $[x_s^{k-1}, x_t^{k-1}]$ with respect to $\bar{x}^k$. Calculate $f(x_{k+1})$ and compare it with $f(\bar{x}^k)$.

(b) Use the elimination scheme for unimodal function to discard $1 - \tau$ of the interval of uncertainty obtained at iteration $k - 1$.

(c) Let $[x_s^k, x_t^k] :=$ the interval of uncertainty obtained at iteration k.

(d) Let $\bar{x}^{k+1} :=$ the test point remaining inside $[x_s^k, x_t^k]$.

Step 4. Output $[x_s^N, x_t^N]$ as the final interval of uncertainty.

Example We consider the following problem:

$$\begin{aligned} \min \quad & f(x) = x^2 + 2x \\ s.t. \quad & -3 \leq x \leq 5. \end{aligned}$$

We use the Golden Section method until the length of the uncertain interval is at most 0.2. In iteration k, let x_s^k and x_t^k be respectively the left and right endpoint of the search interval, and let $\hat{x}^k$ and $\bar{x}^k$ be respectively the left and right test point. The first two test points are chosen as

$$\hat{x}^1 = -3 + 0.382 \times 8 = 0.056, \qquad \bar{x}^1 = -3 + 0.618 \times 8 = 1.944.$$

The computations are summarized in the table below. After eight iterations involving nine function evaluations, the search interval becomes [-1.112, -0.936] which meets the accuracy requirement, so that the minimum point can be estimated to be the midpoint -1.024. Note that the true minimum point is -1.0.

Table: Summary of Computations for the Golden Section Method

Iteration k	x_s^k	x_t^k	$\hat{x}^k$	$\bar{x}^k$	$f(\hat{x}^k)$	$f(\bar{x}^k)$
1	-3.000	5.000	0.056	1.944	0.115*	7.667*
2	-3.000	1.944	-1.112	0.056	-0.987*	0.115
3	-3.000	0.056	-1.832	-1.112	-0.308*	-0.987
4	-1.832	0.056	-1.112	-0.664	-0.987	-0.887*
5	-1.832	-0.664	-1.384	-1.112	-0.853*	-0.987
6	-1.384	-0.664	-1.112	-0.936	-0.987	-0.996*
7	-1.112	-0.664	-0.936	-0.840	-0.996	-0.974*
8	-1.112	-0.840	-1.016	-0.936	-1.000*	-0.996
9	-1.112	-0.936				

* — this function value should be calculated in the iteration

3.2.4 Comparison of the Two Derivative Free Methods

Suppose the length of initial interval of uncertainty $L_0 = 1$ and after N iterations, the length of the final interval is L_N. If we take the mid-point of the final interval as an approximate minimum point, then the maximum error E should be no more than the half-length of the final interval: $E \leq \frac{L_N}{2}$. In the following table we give the number of function evaluations n required to reach different accuracies for the two methods. We see that the Golden Section method apparently outperforms the Exhaustive method.

Method	$E \leq 0.1$	$E \leq 0.01$	$E \leq 0.001$
Exhaustive Search	$n \geq 9$	$n \geq 99$	$n \geq 999$
Golden Section Search	$n \geq 5$	$n \geq 10$	$n \geq 14$

n — the number of function evaluations.

3.3 A One-Dimensional Search Method that Uses First Order Derivatives - Bisection Method

We want to find

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in [x_s, x_t] \end{array}$$

Idea

- We turn to find a stationary point: $f'(x^*) = 0$.
- Assume that $f \in C^1[x_s, x_t]$.
- Let $g(x) = f'(x)$. Then $g \in C^0[x_s, x_t]$.
- **Intermediate Value Theorem.** If $g \in C^0[x_s, x_t]$ and K is any number between $g(x_s)$ and $g(x_t)$, then there exists c in (x_s, x_t) for which $g(c) = K$.

- Suppose that $g(x_s)g(x_t) < 0$. Then, there exists x^* in (x_s, x_t) such that $g(x^*) = f'(x^*) = 0$.
- To begin, set $[x_s^0, x_t^0] := [x_s, x_t]$ and let x^1 be the mid-point of $[x_s^0, x_t^0]$.
- If $g(x^1) = 0$, then set $x^* := x^1$ and stop. If not, then $g(x^1)$ has a different sign from either $g(x_s^0)$ or $g(x_t^0)$.
- If $g(x^1)$ and $g(x_s^0)$ have different signs, then $x^* \in (x_s^0, x^1)$, and we set $[x_s^1, x_t^1] := [x_s^0, x^1]$. If $g(x^1)$ and $g(x_t^0)$ have different signs, then $x^* \in (x^1, x_t^0)$, and we set $[x_s^1, x_t^1] := [x^1, x_t^0]$.

- We then reapply the procedure to the interval $[x_s^1, x_t^1]$.
- In general, we have

$$L_k = \frac{L_{k-1}}{2} = \cdots = \frac{L_0}{2^k}.$$

The length of the final interval of uncertainty is

$$L_N = \frac{L_0}{2^N}.$$

- Totally, we need to calculate $f'(x)$ for $N+2$ times, because to start with, we need to know the values $f'(x_s)$ and $f'(x_t)$, and then in each iteration, we need to calculate derivative $f'(x)$ at the middle point of the search interval.

Algorithm

Step 1. Input and Initialization:

(a) Input $[x_s^0, x_t^0]$ = the initial interval of uncertainty.

(b) Calculate $f'(x_s^0)$ and $f'(x_t^0)$. If $f'(x_s^0)f'(x_t^0) > 0$, then ERROR.

(c) Input N = the total number of iterations required.

(d) Let $L_0 := x_t^0 - x_s^0$.

(In (b), ERROR means a correct search interval has not been located yet so that we cannot start using the method.)

Step 2. For $k = 1, 2, \ldots, N$, do the following:

(a) Let $x^k := \frac{1}{2}(x_s^{k-1} + x_t^{k-1})$ and calculate $f'(x^k)$.

(b) If $f'(x^k) = 0$, then output $x^* = x^k$ and STOP.

(c) If $f'(x^k)f'(x_s^{k-1}) < 0$, then let $[x_s^k, x_t^k] := [x_s^{k-1}, x^k]$, else let $[x_s^k, x_t^k] := [x^k, x_t^{k-1}]$.

(d) Let $L_k := \frac{1}{2}L_{k-1}$.

Step 3. Output $[x_s^N, x_t^N]$ as the final interval of uncertainty which has length L_N.

3.4.1 Newton's Method

Idea

- Assume that $f \in C^2[x_s, x_t]$ is a convex function (if f is not convex, the method can find only a stationary point, not a minimum point).
- Assume that an initial point $x^1 \in [x_s, x_t]$ is given and is sufficiently close to the minimum solution x^* with $f'(x^*) = 0$ and $f''(x^*) > 0$.
- Assume that $f(x^k)$, $f'(x^k)$ and $f''(x^k)$ are known at point x^k in iteration k $(k = 1, 2, \ldots)$.

- Construct a quadratic function which at x^k agrees with f up to second derivatives, i.e.,

$$f(x) \approx q(x) = f(x^k) + f'(x^k)(x - x^k) + \frac{1}{2} f''(x^k)(x - x^k)^2.$$

(It is easy to see that $q(x^k) = f(x^k), q'(x^k) = f'(x^k)$, and $q''(x^k) = f''(x^k)$).

- Let the next guess of the minimum point, x^{k+1}, be the minimum point of $q(x)$, i.e.,

$$q'(x) = 0 \Longrightarrow f'(x^k) + f''(x^k)(x - x^k) = 0 \Longrightarrow x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}.$$

- The process repeats until $|f'(x^{k+1})| < \varepsilon$ or other termination criterion is satisfied.

In fact Newton's method can be used to solve equation $g(x) = 0$ with the formula:

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}.$$

In a minimization problem, we replace $g(x)$ by $f'(x)$.

Advantage

- Fast if x^1 is near x^*. (need fewer iterations than the bisection method does)

Disadvantage

- Need second-order derivatives.
- The sequence $\{x^k\}$ may not converge to the solution x^* if the initial point x^1 is far away from x^*.

Algorithm

Step 1. Input $x^1 =$ the initial point. Set $k = 1$.

Step 2. Repeat the following formula:

$$x^{k+1} := x^k - \frac{f'(x^k)}{f''(x^k)}$$

until $|f'(x^{k+1})| < \varepsilon$ or other stopping criterion is met.

Step 3. Output the last x^{k+1} as x^*.

Geometric Meaning of Newton's Method.

See the figure on page 39. Suppose the current iterative point is x^{k-1}. We approximate the curve $f'(x)$ by its tangent line at the point $(x^{k-1}, f'(x^{k-1}))$. What we need to find is the intersection point x^* of the curve $f'(x)$ with x-axis, and we use the intersection point of the tangent line with x-axis to approximate x^*. The latter is just x^k according to the formula of Newton's method.

3.4.2 Secant Method

Idea

- Assume that $f \in C^2[x_s, x_t]$ is a convex function.
- Assume that two distinct initial points $x^0, x^1 \in [x_s, x_t]$ are given, and they are sufficiently close to the minimum solution x^* with $f'(x^*) = 0$ and $f''(x^*) > 0$.
- Assume that $f'(x^{k-1})$ and $f'(x^k)$ are known at iteration k $(k = 1, 2, \ldots)$.

- Construct a quadratic function:

$$\begin{aligned} q(x) &= f(x^k) + f'(x^k)(x - x^k) \\ &\quad + \frac{1}{2}\left[\frac{f'(x^k) - f'(x^{k-1})}{x^k - x^{k-1}}\right](x - x^k)^2. \end{aligned}$$

Note that $q(x^k) = f(x^k)$ and $q'(x^k) = f'(x^k)$, but $q''(x^k) \neq f''(x^k)$. In fact

$$q''(x^k) = \left[\frac{f'(x^k) - f'(x^{k-1})}{x^k - x^{k-1}}\right] \approx f''(x^k).$$

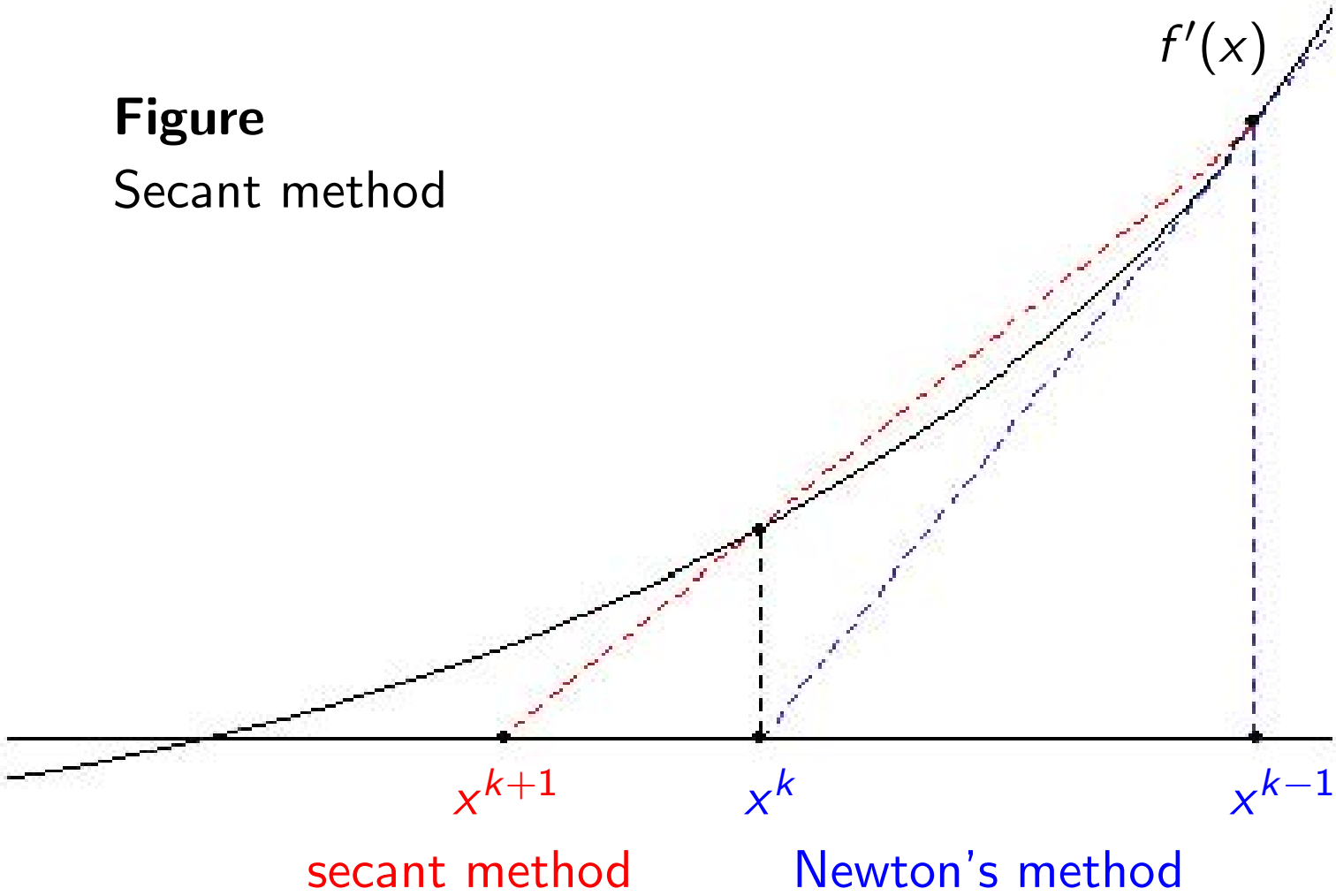
The geometric meaning is that in the graph of $g(x) = f'(x)$, the slope of the tangent line at x^k, i.e. $f''(x^k)$, is approximately replaced by the amount

$$\left[\frac{f'(x^k) - f'(x^{k-1})}{x^k - x^{k-1}}\right],$$

which is in fact the slope of the secant line passing through the two points $(x^{k-1}, f'(x^{k-1}))$ and $(x^k, f'(x^k))$ in the graph of function $f'(x)$. So, we used the secant line to replace the tangent line. That is the reason for the name of the method.

Figure

Secant method



- Let x^{k+1} be the minimum point of $q(x)$, i.e.,

$$x^{k+1} = x^k - \left[\frac{x^k - x^{k-1}}{f'(x^k) - f'(x^{k-1})}\right] f'(x^k).$$

Advantage

- Use first order derivatives only, and do NOT need to use second-order derivatives.

Disadvantage

- May be slower than Newton's Method.

Algorithm

Step 1. Input x^0, x^1 as two initial points.

Step 2. Repeat the following formula for $k = 1, 2, \ldots$

$$x^{k+1} := x^k - \left[\frac{x^{k-1} - x^k}{f'(x^{k-1}) - f'(x^k)}\right] f'(x^k)$$

until $|f'(x^{k+1})| < \varepsilon$ or other stopping criterion is met.

Step 3. Output the last x^{k+1} as x^*.

3.5 Speed of Convergence

- We often need to study how fast an algorithm converges to a minimum solution.
- **Definition.** Let the sequence $\{r_k\}$ converge to r^*. The *order of convergence* of $\{r_k\}$ is defined as the non-negative number p satisfying

$$0 < \lim_{k \to \infty} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^p} = \beta < \infty.$$

- If the sequence has a convergence order p, we have asymptotically,

$$|r_{k+1} - r^*| \approx \beta |r_k - r^*|^p.$$

Note that if a sequence $\{r_k\}$ converges to r^* with a convergence order p', then p' must be unique. That is, $\{r_k\}$ cannot have two different convergence orders.

In fact if $\{r_k\}$ has converges order p', then

$$\lim_{k\to\infty} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^{p'}} = \beta \quad (0 < \beta < \infty).$$

Then for any $p \neq p'$,

$$\begin{aligned} \lim_{k\to\infty} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^{p}} &= \lim_{k\to\infty} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^{p'}} \frac{|r_k - r^*|^{p'}}{|r_k - r^*|^{p}} \\ &= \begin{cases} 0 & \text{if } p < p' \\ \infty & \text{if } p > p' \end{cases} \end{aligned}$$

Therefore, p cannot be another convergence order.

Example 3.5.1:

- The sequence with $r_k = \frac{1}{k}$ converges to zero with order 1.
- The sequence with $r_k = a^k$, where $0 < a < 1$, converges to zero with order 1.
- The sequence with $r_k = a^{2^k}$, where $0 < a < 1$, converges to zero with order two.

Remark 3.5.1 Some convergent sequences may not have a convergence order. For example, in the above example, we know that

- sequence $r_k = \frac{1}{k}$ converges to 0 with order 1;
- sequence $r'_k = a^{2^k}$ $(0 < a < 1)$ converges to 0 with order 2.

Now if we mix the two sequences as

$$S = \{r_1, r'_1, r_2, r'_2, r_3, r'_3, \ldots\ldots\},$$

then the resulting sequence has no convergence order. In fact let the sequence be $S = \{s_1, s_2, s_3, \ldots\}$, then

$$s_{2k} = r'_k, \quad k = 1, 2, 3 \ldots$$

$$s_{2k-1} = r_k, \quad k = 1, 2, 3 \ldots$$

and

$$\lim_{k\to\infty} \frac{|s_{2k} - 0|}{|s_{2k-1} - 0|^p} = \lim_{k\to\infty} \frac{|r'_k|}{|r_k|^p} = \lim_{k\to\infty} \frac{a^{2^k}}{(\frac{1}{k})^p} = 0, \quad \forall p > 0,$$

$$\lim_{k\to\infty} \frac{|s_{2k+1} - 0|}{|s_{2k} - 0|^p} = \lim_{k\to\infty} \frac{|r_{k+1}|}{|r'_k|^p} = \lim_{k\to\infty} \frac{\frac{1}{k+1}}{(a^{2^k})^p} = \infty, \quad \forall p > 0.$$

Therefore, for any $p > 0$,

$$\lim_{k\to\infty} \frac{|s_{k+1} - 0|}{|s_k - 0|^p}$$

does not exist.

The following three convergence speeds are often met:

1. if $p = 1$ and $0 < \beta < 1$, then we say that r_k converges to r^* *linearly*;
2. if $p = 2$ and $0 < \beta$, then we say that r_k converges to r^* *quadratically*;
3. if $1 < p$, or if $p = 1$ and $\beta = 0$, then we say that r_k converges to r^* *superlinearly*.

Linear rate is a relatively slow rate. For a linearly convergent sequence $\{r_k\}$, when k is large, the errors are approximately

$$|r_{k+1} - r^*| \approx \beta |r_k - r^*|.$$

Especially, if β is close to 1, $\{r_k\}$ converges very slowly.

Quadratic convergence is very quick. We know that the third example above has order 2. If $a = \frac{1}{10}$, then the sequence $r_k = a^{2^k}$ becomes

$$\frac{1}{10^2}, \frac{1}{10^4}, \frac{1}{10^8}, \frac{1}{10^{16}}, \ldots$$

which tends to 0 very quickly.

For superlinear convergence, in the first case in the definition, as $p > 1$,

$$\frac{|r_{k+1} - r^*|}{|r_k - r^*|} = \frac{|r_{k+1} - r^*|}{|r_k - r^*|^p} \cdot |r_k - r^*|^{p-1} \to \beta \cdot 0 = 0,$$

which becomes the second case. Hence we can directly define superlinear convergence as

$$\lim_{k \to \infty} \frac{|r_{k+1} - r^*|}{|r_k - r^*|} = 0.$$

Superlinear convergence is considered as a quick convergence, even though it may not converge as fast as quadratic convergence.

Example 3.5.2 The sequence $\{r_k\}$ defined by $r_1 = 4$ and

$$r_{k+1} = 1 + (r_k - 1)/2^k, \quad \text{for } k = 1, 2, \ldots$$

i.e.,

$$\{r_k\} = 4, 2.5, 1.375, 1.0469, \ldots$$

meets the condition:

$$\lim_{k \to \infty} \frac{|r_{k+1} - 1|}{|r_k - 1|} = \lim_{k \to \infty} \frac{1}{2^k} = 0.$$

Hence $\{r_k\}$ converges to 1 superlinearly.

Remark 3.5.2 The sequence $\{r_k\}$ in Example 3.5.2 converges to 1 superlinearly, but it does not have a definite convergence order.

In fact we have seen that when $p = 1$,

$$\lim_{k\to\infty} \frac{|r_{k+1} - 1|}{|r_k - 1|^p} = \lim_{k\to\infty} \frac{r_{k+1} - 1}{r_k - 1} = 0.$$

As the limit is 0, $p = 1$ is not the convergence order. It seems that the convergence order is higher than 1.

But this is not true. We can prove that for any $p > 1$,

$$\lim_{k\to\infty} \frac{|r_{k+1} - 1|}{|r_k - 1|^p} = \infty. \tag{8}$$

So, the sequence also cannot have a convergence order $p > 1$.

We now prove (8). The sequence satisfies:

$$r_2 - 1 = \frac{1}{2}(r_1 - 1) = \frac{1}{2} \cdot 3,$$

$$r_3 - 1 = \frac{1}{2^2}(r_2 - 1) = \frac{1}{2} \cdot \frac{1}{2^2} \cdot 3,$$

$$\cdots\cdots$$

In general,

$$r_{k+1} - 1 = \frac{1}{2} \cdot \frac{1}{2^2} \cdots \frac{1}{2^k} \cdot 3.$$

Let

$$\alpha_k = \frac{|r_{k+1} - 1|}{|r_k - 1|^p} = \frac{r_{k+1} - 1}{(r_k - 1)^p} \quad (p > 1).$$

Then we may obtain

$$\alpha_k = \frac{(2 \cdot 2^2 \cdots 2^{k-1})^{p-1}}{2^k} \cdot 3^{1-p}.$$

As

$$2 \cdot 2^2 \cdots 2^{k-1} = 2^{1+2+\cdots+(k-1)} = 2^{\frac{k(k-1)}{2}},$$

$$\alpha_k = 2^{\frac{k(k-1)}{2}(p-1)-k} \cdot 3^{1-p}.$$

Since $p > 1$,

$$\frac{k(k-1)}{2}(p-1) - k = k[\frac{k-1}{2}(p-1) - 1] \to \infty.$$

Therefore,

$$\alpha_k \to \infty.$$

So far we have introduced the concept of convergence order for a sequence of numbers $\{r_k\}$. For a sequence of points (vectors) x^k in R^n converging to a point (vector) x^*, we may define convergence order of $\{x^k\}$ by the convergence order of the sequence $\{\|x^k - x^*\|\}$. For example, we say that x^k converges to x^* superlinearly if $\|x^k - x^*\|$ converges to 0 superlinearly, i.e.,

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0.$$

Later we shall see that **different algorithms may have different convergence orders**. Here for the single-variable Newton's method and secant method, we give their convergence orders.

Theorem 1. Let function f have a continuous second order derivative, and let x^* satisfy $f'(x^*) = 0$, $f''(x^*) \neq 0$. Then, provided x^0 is sufficiently close to x^*, the sequence $\{x^k\}_{k=0}^{\infty}$ generated by Newton's method converges to x^* with an order of convergence at least two.

Theorem 2. Let function f have a continuous second order derivative, and let x^* satisfy $f'(x^*) = 0$, $f''(x^*) \neq 0$. Then, provided x^0 and x^1 are sufficiently close to x^*, the sequence $\{x^k\}_{k=0}^{\infty}$ generated by the secant method converges to x^* with order $\tau_1 \approx 1.618$.