Contents

Chapter 3

Basics of Mathematical Programming

In Chap. 2, we discussed the conditions satisfied by a local minimum point (the required conditions of a local minimum point) and the conditions which guarantee it to be a minimum point (sufficient conditions for a minimum point) under a finite-dimensional vector space setting. No detailed explanation, however, was provided regarding the method (solution) for finding the local minimum point. In this chapter, we would like to address this ensuing matter. The computational formulation associated to such a problem is called an optimization problem or a mathematical programming problem, and active research is being conducted in the academic field referred to as operations research (OR). Here, we will consider algorithms while showing results that are theoretically obtained or ways to deal with the solution of optimization problems. Much of the content covered here is also valid for abstract optimal design problems in Chap. 7. In fact, in Chap. 7 we will see how the same algorithms can be adapted for function spaces.

In this chapter, we assume that the cost functions (objective and constraint functions) have computable gradients and Hessians. However, the real difficulty from the computational point of view is how to evaluate them. We want the reader bears this in mind.

3.1 Problem Setting

Optimal design problems, as seen in Chap. 1, were viewed as optimization problems with equality constraints (state equations) and inequality constraints of the cost functions $f_0(\boldsymbol{\xi}, \boldsymbol{u})$, $\ldots$, $f_m(\boldsymbol{\xi}, \boldsymbol{u})$ defined by the design variable $\boldsymbol{\xi} \in \Xi$ and state variable $\boldsymbol{u} \in U$. In Chap. 2, such a problem was seen as an optimization problem constructed as $f_0(\boldsymbol{\xi}, \boldsymbol{u})$, $\ldots$, $f_m(\boldsymbol{\xi}, \boldsymbol{u})$ with $\boldsymbol{x} = (\boldsymbol{\xi}, \boldsymbol{u}) \in \Xi \times U$ as a design variable.

In this chapter, we shall recall the definitions given in Chap. 1 and

let $\boldsymbol{\xi}$ be the design variable with $f_0(\boldsymbol{\xi}, \boldsymbol{u}(\boldsymbol{\xi}))$, ... , $f_m(\boldsymbol{\xi}, \boldsymbol{u}(\boldsymbol{\xi}))$ denoted by $\tilde{f}_0(\boldsymbol{\xi})$, ... , $\tilde{f}_m(\boldsymbol{\xi})$, respectively. The differential of $\tilde{f}_0(\boldsymbol{\xi})$, ... , $\tilde{f}_m(\boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$ can be obtained via adjoint variable method as seen in Section 2.8. Furthermore, $\tilde{f}_0(\boldsymbol{\xi})$, ... , $\tilde{f}_m(\boldsymbol{\xi})$ are assumed to be non-linear functions. Actually, in the optimal design problem of Chap. 1 (Problem 1.1.4), even if $f_0(\boldsymbol{u})$ is a linear function with respect to $\boldsymbol{u}$, the equality constraint function $\boldsymbol{h}(\boldsymbol{a}, \boldsymbol{u}) = -\boldsymbol{K}(\boldsymbol{a})\boldsymbol{u} + \boldsymbol{p}$ is non-linear with respect to $(\boldsymbol{a}, \boldsymbol{u})$, hence, $\tilde{f}_0(\boldsymbol{a})$ became a non-linear function.

In this chapter, by denoting the design variable $\boldsymbol{\xi} \in \Xi$ as $\boldsymbol{x} \in X = \mathbb{R}^d$, the non-linear functions $\tilde{f}_0$, ... , $\tilde{f}_m$ as f_0, ... , f_m, and the gradient of these with respect to $\boldsymbol{x}$ as $\boldsymbol{g}_0$, ... , $\boldsymbol{g}_m$, respectively, we can consider the following problem which does not include any equality constraints.

Problem 3.1.1 (Non-linear optimization problem) Let $X = \mathbb{R}^d$. Given the functions f_0, ... , $f_m \in C^1(X; \mathbb{R})$, find an element $\boldsymbol{x}$ which satisfies

$$\min_{\boldsymbol{x} \in X} \{ f_0(\boldsymbol{x}) \mid f_1(\boldsymbol{x}) \leq 0, \ldots, f_m(\boldsymbol{x}) \leq 0 \}.$$

□

The structure of this chapter is as follows. In Sect. 3.2, the definitions relating to convergence and the definition of iterative method, which is a basic way to think about the solutions of non-linear optimization problems, will be presented. Then, from Sect. 3.3 to Sect. 3.5, we will look at solutions with respect to unconstrained optimization problems. After that, we will discuss the solutions of optimization problems with inequality constraints (Problem 3.1.1) in Sect. 3.6 and in the rest of the chapter.

3.2 Iterative Method

Given the solutions of non-linear optimization problems, there do not appear to be any methods which allow us to obtain the optimal solution by solving simultaneous linear equations once without any pre-processing. Usually the iterative method shown below is the standard.

Definition 3.2.1 (Iterative method) A method whereby a non-minimum point $\boldsymbol{x}_0 \in X$ is chosen with respect to Problem 3.1.1, and seeking

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g = \boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g \tag{3.2.1}$$

with respect to $k \in \mathbb{N}$, while determining $\boldsymbol{y}_g \in X$ is called an iterative method. Here, $\boldsymbol{y}_g$ is called a search vector and its size $\|\boldsymbol{y}_g\|_X$, a step size. In contrast, $\bar{\boldsymbol{y}}_g$ is a vector providing direction only, and in this book, it is distinguished from a search vector by referring to it as the search direction. It is assumed that the size of $\bar{\boldsymbol{y}}_g$ does not need to be one. $\bar{\epsilon}_g$ is a positive constant for adjusting its size. Moreover, $\boldsymbol{x}_0$ is called the initial point and $\boldsymbol{x}_k$, where $k \in \mathbb{N}$, is called a trial point. □

From this definition, given an algorithm using iterative methods, there is a need to specify the methods for seeking the search direction $\bar{\boldsymbol{y}}_g$ and a method to appropriately determine the step size $\|\boldsymbol{y}_g\|_X$. We will look at these methods in Sect. 3.3 onwards. Moreover, aside from this iterative method, there is a known numerical solution to optimization problems called the direct method. The direct method is used as a collective term for methods which allow solutions to be sought via a finite number of steps. This method, however, will not be discussed in this book since it is mainly used to deal with linear optimization problems.

For the purpose of later discussions, a glossary representing the characteristics and qualities of the iterative method will be defined.

Definition 3.2.2 (Global convergence) An iterative method is said to have global convergence when an initial point is arbitrarily chosen and yet it generates a sequence of iterates that converges to a point for which a necessary condition of optimality holds. □

Definition 3.2.3 (Convergence rate) Let $\boldsymbol{x}$ be a local minimum and $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ be a sequence of iterates obtained through an iterative method. If there exists an index k_0 and a constant $p \in [1, \infty)$ such that the inequality condition

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|_X \leq \|\boldsymbol{x}_k - \boldsymbol{x}\|_X^p$$

holds for each $k \geq k_0$, then p is called the convergence order of the algorithm. Here, when $p = 1$, $r \in (0, 1)$ and when $p > 1$, r is a positive constant. Moreover, when r can be replaced by a number sequence $\{r_k\}_{k\in\mathbb{N}}$ which converges to zero, the algorithm exhibits a super pth order of convergence. □

3.3 Gradient Method

Let us first examine the gradient method as a procedure to find the search direction $\bar{\boldsymbol{y}}_g$. Here, let us consider choosing one cost function f_i from $i \in \{0, 1, \ldots, m\}$ and obtain the direction in which $\bar{\boldsymbol{y}}_{gi}$ descends. Such a $\bar{\boldsymbol{y}}_{gi}$ will be referred to as the descent direction of f_i.

If in Problem 3.1.1, the minimum point is a point within the admissible set (all inequality constraints become inactive), the descent direction $\bar{\boldsymbol{y}}_{g0}$ of f_0 becomes the search direction $\bar{\boldsymbol{y}}_g$ of Eq. (3.2.1). Moreover, even when obtaining the search direction $\bar{\boldsymbol{y}}_g$ in the case when any of the inequality constraint conditions become active, as will be shown in Sect. 3.7 and beyond, the search direction $\bar{\boldsymbol{y}}_g$ satisfying the inequality constraints can be obtained using the descent direction $\bar{\boldsymbol{y}}_{g0}$ of the objective function f_0 and the descent direction $\bar{\boldsymbol{y}}_{gi}$ of each the active constraint functions f_i. The gradient method is used in this case too.

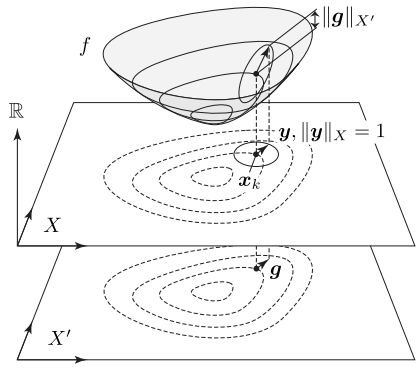


Fig. 3.1: Definition of gradient $\boldsymbol{g}$.

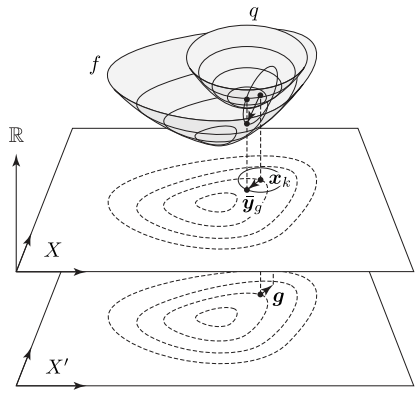


Fig. 3.2: Gradient method.

In this book, from Sect. 3.3 to Sect. 3.5, unconstrained problems will be considered. Here, for simplicity, f_i, $\boldsymbol{g}_i$ and $\bar{\boldsymbol{y}}_{gi}$ are written as f, $\boldsymbol{g}$ and $\bar{\boldsymbol{y}}_g$, respectively.

Before we proceed further, let us define the symbols while referring to Fig. 3.1 and Fig. 3.2. For each $k \in \mathbb{N}$, let $\boldsymbol{x}_k \in X$ be a trial point and $\boldsymbol{g}$ be the gradient of f at $\boldsymbol{x}_k$ not identical to the zero vector $\boldsymbol{0}_X$. We suppose that $\boldsymbol{g}$ is known in advance and then seek the direction $\bar{\boldsymbol{y}}_g \in X$ in which f decreases.

At this point let us confirm the meaning of $\boldsymbol{g}$. Consider the Taylor expansion of f around $\boldsymbol{x}_k$ given by

$$f(\boldsymbol{x}_k + \boldsymbol{y}) = f(\boldsymbol{x}_k) + \boldsymbol{g} \cdot \boldsymbol{y} + o(\|\boldsymbol{y}\|_X). \tag{3.3.1}$$

Here, if the definition of a Fréchet derivative (see Definition 4.5.4 in Chap. 4) is used, then $\boldsymbol{g}$ is an element of the dual space X' (Definition 4.4.5) of X and

the size (norm) of $\boldsymbol{g}$ is defined by

$$\|\boldsymbol{g}\|_{X'} = \max_{\boldsymbol{y}\in X} \frac{\left|\langle \boldsymbol{g}, \boldsymbol{y}\rangle_{X'\times X}\right|}{\|\boldsymbol{y}\|_X} = \max_{\boldsymbol{y}\in X,\ \|\boldsymbol{y}\|_X=1} \left|\langle \boldsymbol{g}, \boldsymbol{y}\rangle_{X'\times X}\right|.$$

If $X = \mathbb{R}^d$, then $X' = \mathbb{R}^d$ and the dual product is given by $\langle \boldsymbol{g}, \boldsymbol{y}\rangle_{X'\times X} = \boldsymbol{g}\cdot\boldsymbol{y}$. Based on this definition, $\|\boldsymbol{g}\|_{X'}$ represents the maximum value of $|\boldsymbol{g}\cdot\boldsymbol{y}|$ over all the direction $\boldsymbol{y} \in X$ such that $\|\boldsymbol{y}\|_X = 1$. Moreover, the direction of $\boldsymbol{g}$ is perpendicular with respect to the contour lines of f. This is because in Eq. (3.3.1), with $\boldsymbol{x}_k + \boldsymbol{y} \in X$ as a point on the contour line and $\boldsymbol{y}$ is taken to be a sufficiently small vector,

$$\boldsymbol{g}\cdot\boldsymbol{y} \approx f(\boldsymbol{x}_k + \boldsymbol{y}) - f(\boldsymbol{x}_k) = 0$$

holds. Figure 3.1 illustrates this relationship.

From these relationships, we infer that $\boldsymbol{g}$ points in the direction such that f increases the most. Hence, from the fact that $X = X'$ holds in a finite-dimensional vector space (Section 4.4.6), if $\bar{\boldsymbol{y}}_g \in X$ is chosen such that

$$\bar{\boldsymbol{y}}_g = -\boldsymbol{g}, \tag{3.3.2}$$

then we get

$$f(\boldsymbol{x}_k + \bar{\boldsymbol{y}}_g) - f(\boldsymbol{x}_k) = -\|\bar{\boldsymbol{y}}_g\|_X^2 + o\left(\|\bar{\boldsymbol{y}}_g\|_X\right).$$

Here, if $\|\bar{\boldsymbol{y}}_g\|_X$ is sufficiently small, f decreases.

Let us generalize this method. The method for obtaining the descent direction $\bar{\boldsymbol{y}}_g \in X$ as the solution to the following problem is called the gradient method.

Problem 3.3.1 (Gradient method) Let $X = \mathbb{R}^d$ and let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ be a positive definite real symmetric matrix (Definition 2.4.5). Let the gradient of f at $\boldsymbol{x}_k \in X$ which is not a local minimum point with respect to $f \in C^1(X;\mathbb{R})$ be $\boldsymbol{g}(\boldsymbol{x}_k) \in X' = \mathbb{R}^d$. In this case, obtain $\bar{\boldsymbol{y}}_g \in X$ which satisfies

$$\bar{\boldsymbol{y}}_g \cdot (\boldsymbol{A}\boldsymbol{y}) = -\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{y} \tag{3.3.3}$$

with respect to an arbitrary $\boldsymbol{y} \in X$. □

Equation (3.3.3) is an expression using the inner product with an arbitrary $\boldsymbol{y} \in X$. This equation is equivalent to obtaining $\bar{\boldsymbol{y}}_g$ via

$$\bar{\boldsymbol{y}}_g = -\boldsymbol{A}^{-1}\boldsymbol{g}. \tag{3.3.4}$$

The reason for using the inner product is so that when defining the gradient method in function space in Chap. 7, it becomes a natural extension of Problem 3.3.1. Moreover, Eq. (3.3.2) is the gradient method in the case when $\boldsymbol{A}$ is the identity matrix $\boldsymbol{I}$. Let us confirm the fact that the solution $\bar{\boldsymbol{y}}_g$ of Problem 3.3.1 reduces f with the following theorem.

Theorem 3.3.2 (Gradient method) The solution $\bar{\boldsymbol{y}}_g$ of Problem 3.3.1 is the descent direction of f at $\boldsymbol{x}_k$. □

Proof Since $\boldsymbol{A}$ is a positive definite symmetric matrix, a positive constant α exists and the inequality

$$\boldsymbol{y} \cdot (\boldsymbol{A}\boldsymbol{y}) \geq \alpha \|\boldsymbol{y}\|_X^2, \quad \boldsymbol{A} = \boldsymbol{A}^\top,$$

holds with respect to an arbitrary $\boldsymbol{y} \in X$. This relationship and Eq. (3.3.3) can be used to establish

$$\begin{aligned} f\left(\boldsymbol{x}_k + \bar{\epsilon}\bar{\boldsymbol{y}}_g\right) - f\left(\boldsymbol{x}_k\right) = \bar{\epsilon}\boldsymbol{g} \cdot \bar{\boldsymbol{y}}_g + o\left(\bar{\epsilon}\right) &= -\bar{\epsilon}\bar{\boldsymbol{y}}_g \cdot \left(\boldsymbol{A}\bar{\boldsymbol{y}}_g\right) + o\left(\bar{\epsilon}\right) \\ &\leq -\bar{\epsilon}\alpha \left\|\bar{\boldsymbol{y}}_g\right\|_X^2 + o\left(\bar{\epsilon}\right) \end{aligned}$$

with respect to the positive constant $\bar{\epsilon}$. Consequently, if $\bar{\epsilon}$ is sufficiently small, then f decreases in value. □

Let us define the descent direction's descent angle as follows.

Definition 3.3.3 (Descent angle) For $\boldsymbol{x}_k \in X$, $\boldsymbol{g} \in X'$ is taken to be the gradient and $\bar{\boldsymbol{y}}_g \in X$ the descent direction. In this case, $\theta \in [0, \pi]$ is defined by

$$\cos\theta = -\frac{\left\langle \boldsymbol{g}, \bar{\boldsymbol{y}}_g \right\rangle_{X' \times X}}{\|\boldsymbol{g}\|_{X'} \left\|\bar{\boldsymbol{y}}_g\right\|_X}$$

is called the descent angle of $\bar{\boldsymbol{y}}_g$ at $\boldsymbol{x}_k$. □

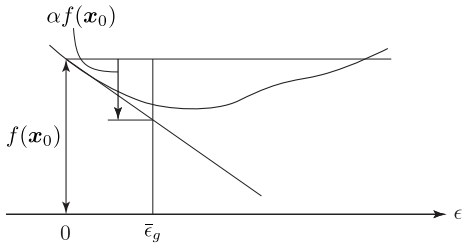
If $\boldsymbol{A}$ is set to be the identity matrix $\boldsymbol{I}$ in the gradient method (Problem 3.3.1), the descent angle θ of $\bar{\boldsymbol{y}}_g$ vanishes. This iterative method is called the maximum descent method. However, it is not necessarily the case that convergence is faster with the maximum descent method. This will become clear through comparison with the conjugate gradient method (Problem 3.4.10) which will be discussed later in this section.

Let us consider the geometric meaning of $\bar{\boldsymbol{y}}_g$ obtained by the gradient method. Problem 3.3.1 is equivalent to seeking $\bar{\boldsymbol{y}}_g \in X$ which satisfies

$$q\left(\bar{\boldsymbol{y}}_g\right) = \min_{\boldsymbol{y} \in X} \left\{ q\left(\boldsymbol{y}\right) = \frac{1}{2}\boldsymbol{y} \cdot (\boldsymbol{A}\boldsymbol{y}) + \boldsymbol{g} \cdot \boldsymbol{y} + f\left(\boldsymbol{x}_k\right) \right\}. \tag{3.3.5}$$

In fact, if the condition $q'\left(\bar{\boldsymbol{y}}_g\right)[\boldsymbol{y}] = 0$ holds true for any $\boldsymbol{y} \in X$, then so does Eq. (3.3.3) and vice versa. Figure 3.2 shows the function q in this case. Here, q is an elliptic paraboloid and its minimum point is $\boldsymbol{x}_k + \bar{\boldsymbol{y}}_g$. The size of $\bar{\boldsymbol{y}}_g$ depends on the choice of $\boldsymbol{A}$. Therefore, if one wants the step size $\left\|\boldsymbol{y}_g\right\|_X = \left\|\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right\|_X$ to be ϵ_g, the following calculation should be carried out. Introduce a positive constant c_a as an adjustment parameter and change Eq. (3.3.4) to

$$\boldsymbol{y}_g = -\left(c_a\boldsymbol{A}\right)^{-1}\boldsymbol{g}. \tag{3.3.6}$$



(Addition)Reduction rate α of objective function.

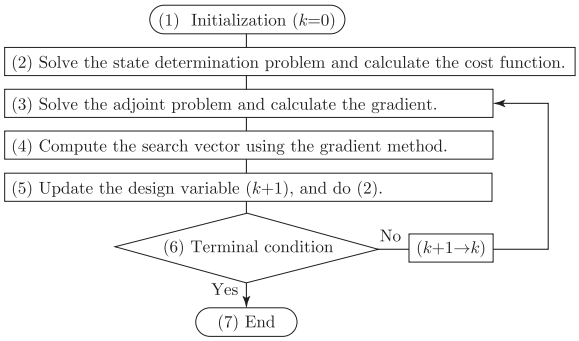


Fig. 3.3: Algorithm of gradient method.

It should be noted here that if c_a is made bigger, the size of $\boldsymbol{y}_g$ becomes smaller. Hence, when the step size ϵ_g and the solution $\bar{\boldsymbol{y}}_g = \bar{\boldsymbol{y}}_g(\boldsymbol{x}_0)$ of the gradient method (Problem 3.3.1) at initial point $\boldsymbol{x}_0$ are given, we have

$$c_a = \frac{\left\|\bar{\boldsymbol{y}}_g\right\|_X}{\epsilon_g}. \tag{3.3.7}$$

Here we assume that c_a is obtained in this way at the initial step ($k = 0$) and its value is used in order to obtain the search vector via Eq. (3.3.6) at the succeeding steps ($k \in \mathbb{N}$) too. Then the step size $\left\|\boldsymbol{y}_{gk}\right\|_X$ roughly takes the size ϵ_g for a while. Moreover, the step size becomes zero when the trial point approaches the point of convergence. In this case, it is equivalent to seeking $\boldsymbol{y}_g \in X$ which satisfies

$$q\left(\boldsymbol{y}_g\right) = \min_{\boldsymbol{y}\in X}\left\{q\left(\boldsymbol{y}\right) = \frac{1}{2}\boldsymbol{y}\cdot\left(c_a\boldsymbol{A}\boldsymbol{y}\right) + \boldsymbol{g}\cdot\boldsymbol{y} + f\left(\boldsymbol{x}_k\right)\right\}. \tag{3.3.8}$$

In the above equation, we emphasize that the magnitude of c_a depends on the choice of the free parameter ϵ_g. So, obtaining an appropriate value for c_a is clearly not straightforward. Nevertheless, if we can determine the step size, then we can use Eq. (3.3.7) to decide for c_a. In the case of domain variations,

discussed in Chap. 9, for instance, the magnitude of a domain variation (step size) is defined by a norm of the strain with respect to the domain variation, such as the square root of the integral of squared strain or the maximum strain. In such a situation, we can imagine that a value of 0.05 for ϵ_g would be a good choice. However, if we do not have any idea about the step size, then we have to consider another way to decide the value of c_a.

One possible way to determine a good choice for c_a is to assume that the objective function reduces at some rate after a domain variation.[1] We illustrate this method as follows. Suppose that the objective function $f(\boldsymbol{x}_0)$ and the gradient $\boldsymbol{g}(\boldsymbol{x}_0)$ at $k = 0$ are given, and a search vector $\bar{\boldsymbol{y}}_g$ is obtained by the gradient method. Also, let us assume that the objective function reduces at a rate of $\alpha \in (0, 1)$ after every domain variation. Then, we have the estimate

$$f\left(\boldsymbol{x}_0 + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) - f(\boldsymbol{x}_0) \approx \alpha f(\boldsymbol{x}_0) \approx \bar{\epsilon}_g \boldsymbol{g}(\boldsymbol{x}_0) \cdot \bar{\boldsymbol{y}}_g.$$

When '$\approx$' is considered '$=$', c_a is given by

$$c_a = \frac{1}{\bar{\epsilon}_g} = \frac{\boldsymbol{g}(\boldsymbol{x}_0) \cdot \bar{\boldsymbol{y}}_g}{\alpha f(\boldsymbol{x}_0)}. \tag{3.3.9}$$

Based on the observations above, let us develop a simple algorithm based on the gradient method. In this chapter, we shall make use of some particular statements when stating the steps of the algorithms. More precisely, with the supposition that optimal design problems may be solved, the following expressions will be used. The phrase "Calculate $f(\boldsymbol{x}_k)$" will be written as "Solve the state determination problem and calculate $f(\boldsymbol{x}_k)$". Furthermore, we will write "Calculate $\boldsymbol{g}(\boldsymbol{x}_k)$" as "Solve the adjoint problem with respect to f and calculate $\boldsymbol{g}(\boldsymbol{x}_k)$". The reason for these is because the calculation becomes like that in an optimal design problem, as explained at the start of Sect. 3.1.

With this background in mind, we now provide examples of algorithms using the gradient method, starting with the simplest one. The adjustment parameter for determining the step size c_a is assumed to be given in advance. Figure 3.3 illustrates an overview of the method.

Algorithm 3.3.4 (Gradient method) In Problem 3.1.1, f_0 is denoted by f and all inequality constraints are assumed to be inactive.

(1) Define the following parameters: initial point $\boldsymbol{x}_0$, positive definite symmetric matrix $\boldsymbol{A}$ ($\boldsymbol{I}$ if there is no particular specification), positive constant for adjusting the step size c_a and positive constant ϵ_0 needed for the convergence check. Set $k = 0$.

(2) Solve the state determination problem and calculate $f(\boldsymbol{x}_k)$.

(3) Solve the adjoint problem with respect to f and calculate $\boldsymbol{g}(\boldsymbol{x}_k)$.

(4) Calculate $\boldsymbol{y}_g$ by Eq. (3.3.6).

[1] Julius Fergy T. Rabago (private communication).

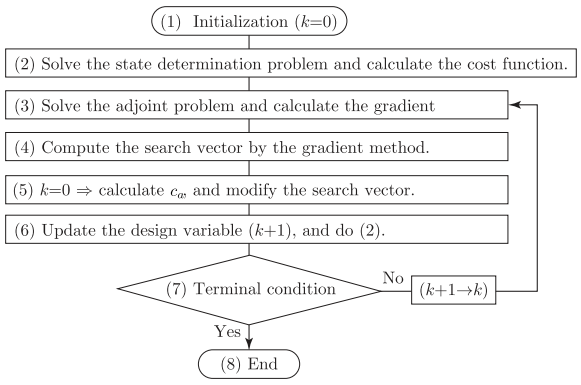


Fig. 3.4: Algorithm of gradient method when the initial value of the step size is given.

(5) Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$. Solve the state determination problem and calculate $f(\boldsymbol{x}_{k+1})$.

(6) Check the final condition $|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| \leq \epsilon_0$.

- Proceed to (7) if the final condition is satisfied.
- Otherwise, substitute $k+1$ into k and return to (3).

(7) Complete the calculation.

□

In Algorithm 3.3.4, $|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| \leq \epsilon_0$ was used as a stopping criterion. Other than this, conditions such as k is not over an upper limit or $\|\boldsymbol{y}_g\|_X \leq \epsilon_0$ in which attention is given to the variation of the design variable can also be utilized.

If one wanted c_a to be determined so that the first step size is a specified ϵ_g (or from the objective function reduce rate α) , the following algorithm can be used. Figure 3.4 shows an overview of its steps.

Algorithm 3.3.5 (Gradient method with initial step size specified)
In Problem 3.1.1, f_0 is denoted by f and all inequality constraints are assumed to be inactive.

(1) Define the following parameters: initial point $\boldsymbol{x}_0$, positive definite symmetric matrix $\boldsymbol{A}$, positive constant for the initial step-size ϵ_g (or the objective function reduce rate α) and positive constant ϵ_0 needed for the convergence check. Let $c_a = 1$ and set $k = 0$.

(2) Solve the state determination problem and calculate $f(\boldsymbol{x}_k)$.

(3) Solve the adjoint problem with respect to f and calculate $\boldsymbol{g}(\boldsymbol{x}_k)$.

(4) Use Eq. (3.3.6) to calculate $\boldsymbol{y}_g$.

(5) When $k = 0$, let $\boldsymbol{y}_g = \bar{\boldsymbol{y}}_g$ and obtain c_a using Eq. (3.3.7). Moreover, substitute $\bar{\boldsymbol{y}}_g / c_a$ into $\boldsymbol{y}_g$.

(6) Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$. Solve the state determination problem and calculate$f(\boldsymbol{x}_{k+1})$.

(7) Check the final condition $|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| \leq \epsilon_0$.

- Proceed to (8) when the terminal condition is satisfied.
- Otherwise, substitute $k+1$ into k and return to (3).

(8) Complete the calculation.

□

3.4 Step Size Criterion

Next let us consider a method for appropriately deciding the step size $\left\|\boldsymbol{y}_g\right\|_X$.

If the search direction $\bar{\boldsymbol{y}}_g$ is already known, the variable in an optimization problem is only $\bar{\epsilon}_g$ of Eq. (3.2.1). Hence, an optimization problem such as the following with $\bar{\epsilon}_g$ taken to be a design variable can be considered to suggest a method for determining the step size $\left\|\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right\|_X$ from its solutions. This method is called the strict line search method.

Problem 3.4.1 (Strict line search method) Let $X = \mathbb{R}^d$. Given $f \in C^1(X;\mathbb{R})$, $\boldsymbol{x}_k \in X$ and $\bar{\boldsymbol{y}}_g \in X$, obtain $\bar{\epsilon}_g$ which satisfies

$$\min_{\bar{\epsilon}_g \in (0,\infty)} f\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right).$$

□

The algorithm for solving Problem 3.4.1 employs methods to solve non-linear equations. For example, a method such as

- Bisection method
- Secant method

can be considered. The bisection method repeats operations for finding the mid-point of the region such as $\bar{\epsilon}_g$ in which f changes from decreasing to increasing. In this case, the convergence order to an exact solution for $\bar{\epsilon}_g$ is one. When using the secant method, it is viewed as a problem obtaining $\bar{\epsilon}_g$ such that the gradient of f with respect to $\bar{\epsilon}_g$ is zero. To solve the problem, the updating equation of the Newton–Raphson method, which will be shown later, is used. However, in the secant method, the gradient of f is replaced with the difference (Practices **3.1** and **3.2**). It is known that the convergence order of this method is the golden ratio $\left(1+\sqrt{5}\right)/2$.

The following results can be obtained regarding the convergence of an exact solution of $\boldsymbol{x}_k$ computed through the strict line search method. Assume that the cost function is a second-order function. Suppose that the search direction is obtained via the maximum descent method. In this case, the step size obtained from the strict line search method is the solution of the following problem.

Problem 3.4.2 (Strict line search of 2nd-order optimization problem) Let $X = \mathbb{R}^d$. Suppose $\boldsymbol{B} \in \mathbb{R}^{d\times d}$ is a positive definite symmetric matrix, $\boldsymbol{b} \in X$ is a given vector, and

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}\cdot(\boldsymbol{B}\boldsymbol{x}) + \boldsymbol{b}\cdot\boldsymbol{x} \tag{3.4.1}$$

is a cost function. Given these assumptions, find $\bar{\boldsymbol{y}}_g \in X$ using the maximum descent method with respect to $\boldsymbol{x}_k \in X$ and obtain $\bar{\epsilon}_g \in (0,\infty)$ which satisfies Problem 3.4.1. □

Answer If $f\left(\boldsymbol{x}_k + \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)$ is written as $\bar{f}(\bar{\epsilon}_g)$, we can write

$$\begin{aligned}\bar{f}(\bar{\epsilon}_g) &= \frac{1}{2}\left(\boldsymbol{x}_k + \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)\cdot\left\{\boldsymbol{B}\left(\boldsymbol{x}_k + \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)\right\} + \boldsymbol{b}\cdot\left(\boldsymbol{x}_k + \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)\\ &= \bar{\epsilon}_g^2\frac{1}{2}\bar{\boldsymbol{y}}_g\cdot\left(\boldsymbol{B}\bar{\boldsymbol{y}}_g\right) + \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\cdot\boldsymbol{g} + f(\boldsymbol{x}_k),\end{aligned}$$

where $\boldsymbol{g} = \boldsymbol{B}\boldsymbol{x}_k + \boldsymbol{b}$ was used in the second equality. In view of the strict line search method, the equation

$$\frac{\mathrm{d}\bar{f}}{\mathrm{d}\bar{\epsilon}_g} = \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\cdot\left(\boldsymbol{B}\bar{\boldsymbol{y}}_g\right) + \bar{\boldsymbol{y}}_g\cdot\boldsymbol{g} = 0$$

yields

$$\bar{\epsilon}_g = -\frac{\bar{\boldsymbol{y}}_g\cdot\boldsymbol{g}}{\bar{\boldsymbol{y}}_g\cdot\left(\boldsymbol{B}\bar{\boldsymbol{y}}_g\right)}.$$

Furthermore, if $\bar{\boldsymbol{y}}_g$ is the solution of the maximum descent method, then one has $\bar{\boldsymbol{y}}_g = -\boldsymbol{g}$, which gives

$$\bar{\epsilon}_g = -\frac{\bar{\boldsymbol{y}}_g\cdot\boldsymbol{g}}{\bar{\boldsymbol{y}}_g\cdot\left(\boldsymbol{B}\bar{\boldsymbol{y}}_g\right)} = \frac{\boldsymbol{g}\cdot\boldsymbol{g}}{\boldsymbol{g}\cdot(\boldsymbol{B}\boldsymbol{g})} = \frac{\boldsymbol{g}\cdot\boldsymbol{g}}{\bar{\boldsymbol{y}}_g\cdot\left(\boldsymbol{B}\bar{\boldsymbol{y}}_g\right)}. \tag{3.4.2}$$

□

In this way, the strict line search method can be used to provide the following results regarding the convergence when the iterative method is repeated while seeking $\bar{\epsilon}_g$.

Theorem 3.4.3 (Convergence of the strict line search method) The sequence of iterates $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ formed by the iterative method using the solution to Problem 3.4.2, $\bar{\boldsymbol{y}}_g \in X$, and $\bar{\epsilon}_g$ satisfies

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|_{\boldsymbol{B}} \le \left|\frac{\lambda_d - \lambda_1}{\lambda_1 + \lambda_d}\right| \|\boldsymbol{x}_k - \boldsymbol{x}\|_{\boldsymbol{B}},$$

where $\boldsymbol{x}$ is a local minimum point, λ_1 and λ_d denote the minimum and maximum eigenvalues, respectively, and $\|\boldsymbol{x}\|_{\boldsymbol{B}} = \sqrt{\boldsymbol{x} \cdot (\boldsymbol{B}\boldsymbol{x})}$. □

Proof The objective function in Problem 3.4.2 can be written as

$$f(\boldsymbol{x}) = \frac{1}{2}\left(\boldsymbol{x} + \boldsymbol{B}^{-1}\boldsymbol{b}\right) \cdot \left\{\boldsymbol{B}\left(\boldsymbol{x} + \boldsymbol{B}^{-1}\boldsymbol{b}\right)\right\} - \frac{1}{2}\boldsymbol{b} \cdot \left(\boldsymbol{B}^{-1}\boldsymbol{b}\right).$$

Observe that even if $\boldsymbol{x} + \boldsymbol{B}^{-1}\boldsymbol{b}$ is replaced by $\boldsymbol{x}$, the evaluation of $\boldsymbol{x}_{k+1} - \boldsymbol{x}$ remains unchanged. Moreover, since the second term on the right-hand side in the above equation is independent of $\boldsymbol{x}$, then even if it is omitted, the evaluation of $\boldsymbol{x}_{k+1} - \boldsymbol{x}$ does not change. Therefore, it suffices to consider the problem of finding the minimum point of

$$\bar{f}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x} \cdot (\boldsymbol{B}\boldsymbol{x}).$$

When $\boldsymbol{g}_k = \boldsymbol{g}(\boldsymbol{x}_k) = \boldsymbol{B}\boldsymbol{x}_k$, the maximum descent method can be used to obtain $\bar{\boldsymbol{y}}_k = -\boldsymbol{g}_k$. Furthermore, as a result of the strict line search method, Eq. (3.4.2) can be used to obtain

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{\boldsymbol{g}_k \cdot \boldsymbol{g}_k}{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{g}_k)}\bar{\boldsymbol{y}}_k$$

to form the point sequence. In this case,

$$\begin{aligned}
\bar{f}(\boldsymbol{x}_{k+1}) &= \frac{1}{2}\left(\boldsymbol{x}_k - \frac{\boldsymbol{g}_k \cdot \boldsymbol{g}_k}{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{g}_k)}\boldsymbol{g}_k\right) \cdot \left\{\boldsymbol{B}\left(\boldsymbol{x}_k - \frac{\boldsymbol{g}_k \cdot \boldsymbol{g}_k}{\boldsymbol{g}_k \cdot \boldsymbol{B}\boldsymbol{g}_k}\boldsymbol{g}_k\right)\right\} \\
&= \frac{1}{2}\left[\boldsymbol{x}_k \cdot (\boldsymbol{B}\boldsymbol{x}_k) - \frac{2(\boldsymbol{g}_k \cdot \boldsymbol{g}_k)\{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{x}_k)\} - (\boldsymbol{g}_k \cdot \boldsymbol{g}_k)^2}{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{g}_k)}\right] \\
&= \frac{1}{2}\left\{\boldsymbol{x}_k \cdot (\boldsymbol{B}\boldsymbol{x}_k) - \frac{(\boldsymbol{g}_k \cdot \boldsymbol{g}_k)^2}{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{g}_k)}\right\} \\
&= \frac{1}{2}\boldsymbol{x}_k \cdot (\boldsymbol{B}\boldsymbol{x}_k)\left(1 - \frac{(\boldsymbol{g}_k \cdot \boldsymbol{g}_k)^2}{\{\boldsymbol{x}_k \cdot (\boldsymbol{B}\boldsymbol{x}_k)\}\{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{g}_k)\}}\right) \\
&= \left(1 - \frac{(\boldsymbol{g}_k \cdot \boldsymbol{g}_k)^2}{\{\boldsymbol{g}_k \cdot (\boldsymbol{B}^{-1}\boldsymbol{g}_k)\}\{\boldsymbol{g}_k \cdot (\boldsymbol{B}\boldsymbol{g}_k)\}}\right)\bar{f}(\boldsymbol{x}_k)
\end{aligned}$$

is established. Since $\boldsymbol{B}$ is positive definite, we can apply Kantorovich's inequality to obtain

$$\frac{4\lambda_1\lambda_d}{(\lambda_1 + \lambda_d)^2} \leq \frac{(\boldsymbol{y} \cdot \boldsymbol{y})^2}{\{\boldsymbol{y} \cdot (\boldsymbol{B}^{-1}\boldsymbol{y})\}\{\boldsymbol{y} \cdot (\boldsymbol{B}\boldsymbol{y})\}},$$

for any $\boldsymbol{y} \in X$. Therefore,

$$\bar{f}(\boldsymbol{x}_{k+1}) \leq \left(1 - \frac{4\lambda_1\lambda_d}{(\lambda_1 + \lambda_d)^2}\right)\bar{f}(\boldsymbol{x}_k) = \left(\frac{\lambda_d - \lambda_1}{\lambda_1 + \lambda_d}\right)^2 \bar{f}(\boldsymbol{x}_k),$$

and thus, the desired result. □

Let us consider the characteristics of the strict line search method with the above results in mind. The strict line search method requires the minimization problem with only the step size as a design variable to be solved accurately.

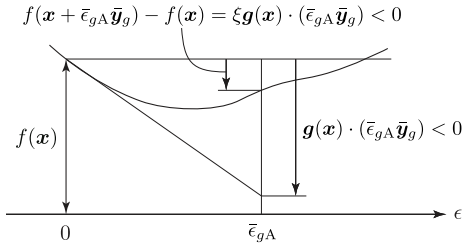


Fig. 3.5: Armijo criterion.

To do this, iterative algorithms such as the bisection method or the secant method are of great practical importance, and thus become necessary. For problems where the calculation of the gradient is rather more difficult compared to the calculation of the cost function with respect to the design variable, it is considered that an effective algorithm can be formulated such that the calculation of the gradient is unnecessary, like the bisection method. However, once the design variable moves in the search direction, the gradient changes and the search direction determined by the gradient method also changes. Even in such situations, it is considered that it is not necessarily a good idea to seek the minimum point accurately using the old search direction. In particular, in the case when using an algorithm in which there is a need for the recalculation of the gradient after $\bar{\epsilon}_g$ is updated, as with the secant method (Practice **3.2**), it is considered that updating the search direction via the gradient method would improve convergence rather than just continuing to use the old search direction.

In the sequel, we shall examine a method focusing on the range over which the non-linearity and gradient of the cost function is effective without worrying about whether it is strictly the case. The criterion shown below provides the upper and lower limit of the step size. With respect to the upper limit of the step size $\left\|\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right\|_X$, conditions such as the one that follows are known [**?**].

Definition 3.4.4 (Armijo criterion) Suppose $\boldsymbol{g}(\boldsymbol{x}_k)$ is the gradient of $f(\boldsymbol{x}_k)$, $\bar{\boldsymbol{y}}_g$ is the search direction and $\xi \in (0,1)$ is the parameter adjusting the upper limit of the step size. If

$$f\left(\boldsymbol{x}_k+\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)-f\left(\boldsymbol{x}_k\right)\leq\xi\boldsymbol{g}\left(\boldsymbol{x}_k\right)\cdot\left(\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)<0 \tag{3.4.3}$$

holds for any $\bar{\epsilon}_g>0$, $\bar{\epsilon}_g$ satisfies the Armijo criterion. □

If the upper limit of $\bar{\epsilon}_g$ satisfying the Armijo criterion is written as $\bar{\epsilon}_{g\mathrm{A}}$, a relationship such as shown in Fig. 3.5 is established. The left-hand side of Eq. (3.4.3) takes a negative value by which the non-linear function f actually reduces when $\boldsymbol{x}$ fluctuates from $\boldsymbol{x}_k$ by just $\bar{\epsilon}_g\bar{\boldsymbol{y}}_g$. In contrast, the quantity $\boldsymbol{g}\left(\boldsymbol{x}_k\right)\cdot\left(\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)$ on the right-hand side admits a negative value when the reduction in f is estimated using the gradient. Equality holds when $\bar{\epsilon}_g$ is sufficiently small. However, the case is different when $\bar{\epsilon}_g$ is of a certain size. Instances when

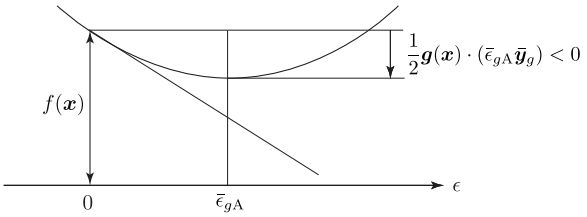


Fig. 3.6: Armijo criterion with respect to a second-order function.

$\xi \in (0,1)$ provides the ratio by which this difference is permitted. Making ξ close to unity represents the fact that their difference is not allowed and making ξ close to zero represents that their difference is permitted. Therefore, the Armijo criterion in effect provides the condition for deciding the step size at a level such that the estimated value using the gradient of f is not too far away from the actual value of reduction. It must be noted that ξ is restricted on the unit interval $(0,1)$ since the Armijo criterion actually fails when $\xi > 1$.

Moreover, the following results can be used as a benchmark for ξ. If $f(\boldsymbol{x})$ is a second-order function and the upper limit of the Armijo criterion when $\xi = 1/2$ is $\bar{\epsilon}_{g\mathrm{A}}$, $\boldsymbol{x}_k + \bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g$ becomes the minimum point of f, see Fig. 3.6. In fact,

$$f\left(\boldsymbol{x}_k + \bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right) - f\left(\boldsymbol{x}_k\right) = \boldsymbol{g}\left(\boldsymbol{x}_k\right)\cdot\left(\bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right) + \frac{1}{2}\left(\bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right)\cdot\left(\boldsymbol{B}\left(\bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right)\right) \tag{3.4.4}$$

is established with respect to the second-order function of Eq. (3.4.1). Here, if $\boldsymbol{x}_k + \bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g$ is a local minimum point, the Taylor expansion of $\boldsymbol{g}\left(\boldsymbol{x}_k\right)$ is given by

$$\boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right) = \boldsymbol{g}\left(\boldsymbol{x}_k\right) + \boldsymbol{B}\left(\bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right) = \boldsymbol{0}_{X'}. \tag{3.4.5}$$

If Eq. (3.4.5) is substituted into Eq. (3.4.4), then one obtains

$$f\left(\boldsymbol{x}_k + \bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right) - f\left(\boldsymbol{x}_k\right) = \frac{1}{2}\boldsymbol{g}\left(\boldsymbol{x}_k\right)\cdot\left(\bar{\epsilon}_{g\mathrm{A}}\bar{\boldsymbol{y}}_g\right).$$

On the other hand, conditions such as the following are known for providing the lower limit for the step size $\left\|\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right\|_X$ [**?**].[2]

Definition 3.4.5 (Wolfe criterion) Let $\boldsymbol{g}\left(\boldsymbol{x}_k\right)$ be the gradient of $f\left(\boldsymbol{x}_k\right)$, $\bar{\boldsymbol{y}}_g$ the search direction, $\xi \in (0,1)$ the parameter used in the Armijo criterion, $\mu \in (0,1)$ the parameter which adjusts the lower limit of the step size and suppose that $0 < \xi < \mu < 1$ is satisfied. In this case, if

$$\mu\boldsymbol{g}\left(\boldsymbol{x}_k\right)\cdot\left(\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right) \leq \boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right)\cdot\left(\bar{\epsilon}_g\bar{\boldsymbol{y}}_g\right) < 0 \tag{3.4.6}$$

holds with respect to $\bar{\epsilon}_g > 0$, $\bar{\epsilon}_g$ is said to satisfy the Wolfe criterion. □

[2] in the literature, the Armijo criterion is in fact a special case of the Wolfe criterion, but in this book, only the condition which gives the lower limit of $\bar{\epsilon}_g$ is referred to as the Wolfe criterion.

If the lower limit value of $\bar{\epsilon}_g$ which satisfies the Wolfe criterion is written as $\bar{\epsilon}_{g\mathrm{W}}$, a geometric relationship such as the one shown in Fig. 3.7 is established. The term $\boldsymbol{g}(\boldsymbol{x}_k)$ on the left-hand side of Eq. (3.4.6) represents the gradient of f at $\boldsymbol{x}_k$. On the other hand, the expression $\boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right)$ on the right-hand side represents the gradient of f obtained when moving $\boldsymbol{x}$ from $\boldsymbol{x}_k$ to $\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g$. The following observations regarding the Wolfe criterion are established:

(1) If the condition $\boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \leq \boldsymbol{g}(\boldsymbol{x}_k) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) < 0$ holds for some $\bar{\epsilon}_g > 0$, then there is no $\bar{\epsilon}_g > 0$ such that Eq. (3.4.6) holds. This condition expresses the fact that when $\boldsymbol{x}$ moves in the direction of $\bar{\boldsymbol{y}}_g$, the negative gradient which would reduce f becomes an even greater negative gradient. It shows that in such a case there is no need to provide a lower limit to the step size.

(2) If, for some $\bar{\epsilon}_g > 0$, $\boldsymbol{g}(\boldsymbol{x}_k) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) < \boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) < 0$ holds, $\bar{\epsilon}_g > 0$ exists such that Eq. (3.4.6) holds. This condition shows, in contrast to (1) above, that the gradient decreases when $\boldsymbol{x}$ moves in the direction of $\bar{\boldsymbol{y}}_g$. If μ is made smaller than one in Eq. (3.4.6), the lower limit $\bar{\epsilon}_{g\mathrm{W}}$ of $\bar{\epsilon}_g$ becomes bigger.

Hence, the Wolfe criterion is a condition which ideally requires the step size to be large enough such that the validity of the gradient is lost to around the proportion of μ.

Meanwhile, the requirement $\xi < \mu$ is based on the following observations. The Taylor expansion of f about $\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g$ is written as

$$f(\boldsymbol{x}_k) = f\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) - \boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) + o(\bar{\epsilon}_g).$$

In this case, if the Wolfe criterion is satisfied, then the following relations hold:

$$\begin{aligned} \mu \boldsymbol{g}(\boldsymbol{x}_k) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) - o(\bar{\epsilon}_g) &\leq \boldsymbol{g}\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) - o(\bar{\epsilon}_g) \\ &= f\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) - f(\boldsymbol{x}_k). \end{aligned}$$

On the other hand, if the Armijo criterion is satisfied, then we have that

$$f\left(\boldsymbol{x}_k + \bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) - f(\boldsymbol{x}_k) \leq \xi \boldsymbol{g}(\boldsymbol{x}_k) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right).$$

Hence, if both conditions are satisfied, then the following requirement must hold:

$$(\mu - \xi)\, \boldsymbol{g}(\boldsymbol{x}_k) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \leq o(\bar{\epsilon}_g).$$

Here, the fact that $\boldsymbol{g}(\boldsymbol{x}_k) \cdot \left(\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right) \leq 0$ implies that if the right-hand side is positive, or else the absolute value is sufficiently small, then the inequality holds when $\xi < \mu$.

An example of an algorithm in which the step size is controlled so that the Armijo criterion and the Wolfe criterion are satisfied is shown below. Figure 3.8 provides an overview of the steps in the algorithm.

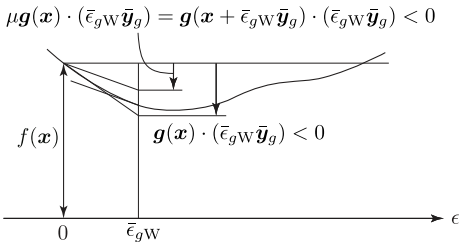


Fig. 3.7: Wolfe criterion.

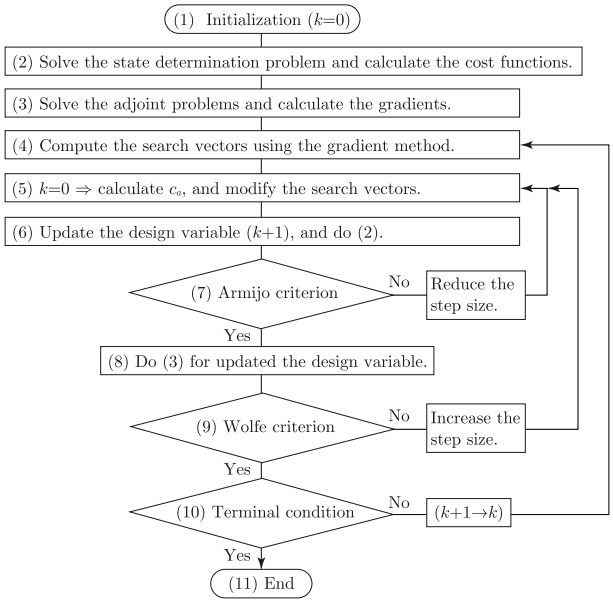


Fig. 3.8: Gradient method algorithm using Armijo criterion and Wolfe criterion.

Algorithm 3.4.6 (Armijo criterion and Wolfe criterion)
Consider Problem 3.1.1. Let f_0 be f and all inequality constraints be inactive.

(1) Define the following parameters: the initial point $\boldsymbol{x}_0$, positive definite symmetric matrix $\boldsymbol{A}$, step size ϵ_g, convergence check value ϵ_0, parameters ξ and μ $(0 < \xi < \mu < 1)$ used in the Armijo criterion and the Wolfe criterion, respectively. Let $c_a = 1$ and set $k = 0$.

(2) Solve the state determination problem and calculate $f(\boldsymbol{x}_k)$.

(3) Solve the adjoint problem with respect to f and calculate $\boldsymbol{g}(\boldsymbol{x}_k)$.

(4) Use Eq. (3.3.6) to calculate $\boldsymbol{y}_g$.

(5) When $k = 0$, let $\boldsymbol{y}_g = \bar{\boldsymbol{y}}_g$ and use Eq. (3.3.7) to obtain c_a. Moreover, substitute $\bar{\boldsymbol{y}}_g / c_a$ into $\boldsymbol{y}_g$.

(6) Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$. Solve the state determination problem and calculate $f(\boldsymbol{x}_{k+1})$.

(7) Check the Armijo criterion (Eq. (3.4.3)).

- Proceed to the next step if satisfied.
- Otherwise, suppose $\alpha > 1$, substitute αc_a into c_a, and $\alpha \boldsymbol{y}_g$ into $\boldsymbol{y}_g$, then return to (5).

(8) Calculate $\boldsymbol{g}(\boldsymbol{x}_{k+1})$.

(9) Check the Wolfe criterion (Eq. (3.4.6)).

- Proceed to the next step if satisfied.
- Otherwise, suppose $\beta < (0, 1)$, substitute βc_a into c_a and substitute $\beta \boldsymbol{y}_g$ into $\boldsymbol{y}_g$, then return to (5).

(10) Check termination condition $|f_0(\boldsymbol{x}_{k+1}) - f_0(\boldsymbol{x}_k)| \leq \epsilon_0$.

- Proceed to (11) when the stopping criterion is satisfied.
- Otherwise, substitute $k + 1$ into k and return to (4).

(11) Complete the calculation.

□

Concerning the sequence of iterates obtained through the algorithms in which the step size is restricted so that the Armijo criterion and the Wolfe criterion are satisfied, the following results relating to global convergence can be obtained.

Theorem 3.4.7 (Global convergence theorem) Let $X = \mathbb{R}^d$. Suppose that the function $f : X \to \mathbb{R}$ has a lower bound, is differentiable in the neighborhood $L = \{\boldsymbol{x} \in X \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ of the level set at $\boldsymbol{x}_0 \in X$ and the gradient $\boldsymbol{g}$ is Lipschitz continuous (Definition 4.3.1) in L. Let the search vector at $\boldsymbol{x}_k$ be $\boldsymbol{y}_{gk}$ and suppose $\boldsymbol{y}_{gk}$ satisfies $\cos\theta_k > 0$ with respect to descent angle θ_k. Furthermore, suppose that the step size $\left\|\bar{\epsilon}_g \bar{\boldsymbol{y}}_g\right\|_X$ satisfies the Armijo criterion and the Wolfe criterion. Under these assumptions, the sequence of iterates $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ generated by the gradient method satisfies

$$\sum_{k\in\mathbb{N}} \|\boldsymbol{g}(\boldsymbol{x}_k)\|_{X'}^2 \cos^2\theta_k < \infty. \tag{3.4.7}$$

□

Proof From the Armijo criterion, we know $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ is in the neighborhood of L. Moreover, the Wolfe criterion implies that the inequality

$$(\mu-1)\,\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{y}_g \le (\boldsymbol{g}(\boldsymbol{x}_{k+1})-\boldsymbol{g}(\boldsymbol{x}_k))\cdot\boldsymbol{y}_g$$

holds. Furthermore, since $\boldsymbol{g}$ is Lipschitz continuous, we have

$$(\boldsymbol{g}(\boldsymbol{x}_{k+1})-\boldsymbol{g}(\boldsymbol{x}_k))\cdot\boldsymbol{y}_g \le \beta\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|_X\|\boldsymbol{y}_g\|_X = \bar{\epsilon}_g\beta\|\boldsymbol{y}_g\|_X^2$$

for some some $\beta>0$. From these equations, we get

$$\bar{\epsilon}_g \ge \frac{(\boldsymbol{g}(\boldsymbol{x}_{k+1})-\boldsymbol{g}(\boldsymbol{x}_k))\cdot\boldsymbol{y}_g}{\beta\|\boldsymbol{y}_g\|_X^2} \ge \frac{(\mu-1)\,\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{y}_g}{\beta\|\boldsymbol{y}_g\|_X^2}.$$

Substituting this equation into the Armijo criterion, we obtain

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &\le f(\boldsymbol{x}_k)+\xi\bar{\epsilon}_g\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{y}_g = f(\boldsymbol{x}_k)-\xi\frac{\mu-1}{\beta}\left(\frac{\boldsymbol{g}(\boldsymbol{x}_k)\cdot\boldsymbol{y}_g}{\|\boldsymbol{y}_g\|_X}\right)^2 \\ &= f(\boldsymbol{x}_k)-\xi\frac{\mu-1}{\beta}\|\boldsymbol{g}(\boldsymbol{x}_k)\|_{X'}^2\cos^2\theta_k. \end{aligned}$$

Therefore, we have

$$f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k)-\xi\frac{\mu-1}{\beta}\sum_{k\in\{0,\cdots,m\}}\|\boldsymbol{g}(\boldsymbol{x}_k)\|_{X'}^2\cos^2\theta_k.$$

Since f is bounded below, the the desired result follows. This proves the theorem. □

Equation 3.4.7 is called Zoutendijk condition. If the result of Theorem 3.4.7 and the necessary conditions for an infinite series to converge, $\lim_{k\to\infty}\|\boldsymbol{g}(\boldsymbol{x}_k)\|_{X'}^2\cos^2\theta_k=0$, are used, a result such as the one that follows is obtained.

Corollary 3.4.8 (Global convergence theorem) In addition to the suppositions of Theorem 3.4.7, if $\boldsymbol{y}_g$ is not asymptotic to the direction which crosses $-\boldsymbol{g}(\boldsymbol{x}_k)$ orthogonally, i.e., when $\cos\theta_k>0$, we have that

$$\lim_{k\to\infty}\boldsymbol{g}(\boldsymbol{x}_k)=\boldsymbol{0}_{X'}.$$

□

This result shows that given an appropriate problem setting, if the search direction is obtained via the gradient method and the step size is chosen so that the Armijo criterion and the Wolfe criterion are satisfied, the generated sequence of iterates $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ has global convergence.

For the rest of this section, we shall introduce the conjugate gradient method as an extension of the gradient method. Let us first define the conjugates between vectors as follows.

Definition 3.4.9 (Conjugate) Suppose $\boldsymbol{B} \in \mathbb{R}^{d\times d}$ is a positive definite real symmetric matrix. Let $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in X$. If $\boldsymbol{x} \cdot (\boldsymbol{B}\boldsymbol{y}) = 0$, then $\boldsymbol{x}$ and $\boldsymbol{y}$ are said to be conjugate. □

In view of Problem 3.4.2, the conjugate gradient method is given as follows.

Problem 3.4.10 (Conjugate gradient method) For each $\boldsymbol{x}_0 \in X$, the search direction $\bar{\boldsymbol{y}}_{g0}$ and the parameter $\bar{\epsilon}_{g0}$ which adjusts the step size are obtained through the steepest gradient method and the strict line search method, respectively. For each $k \in \mathbb{N}$, provide a value to $\bar{\boldsymbol{y}}_{g\,k-1}$and obtain $\bar{\boldsymbol{y}}_{gk}$ such that it is conjugate to $\bar{\boldsymbol{y}}_{g\,k-1}$. In addition, find the value of the parameter $\bar{\epsilon}_{gk}$ which adjusts the step size based on the strict line search method. □

Figure 3.9 shows a geometric illustration of the search vector obtained via the conjugate gradient method when $X = \mathbb{R}^2$. By $\bar{\boldsymbol{y}}_{g0}$ and $\bar{\boldsymbol{y}}_{g1}$ being chosen so that they are conjugates, making it a two-dimensional vector space Problem 3.4.2, the minimum point can be achieved by seeking the search vector only twice.

Let us show an example of a conjugate gradient method. Let $\boldsymbol{x}_0 = \boldsymbol{0}_X$. Use the steepest gradient method to set the search direction to be $\bar{\boldsymbol{y}}_{g0} = -\boldsymbol{g}_0 = -\boldsymbol{g}(\boldsymbol{x}_0) = -\boldsymbol{B}\boldsymbol{x}_0 - \boldsymbol{b} = -\boldsymbol{b}$. If, with respect to $k \in \mathbb{N}$, $\bar{\boldsymbol{y}}_{gk}$ and $\boldsymbol{g}_k$ are given, seek

$$\bar{\epsilon}_{gk} = \frac{\bar{\boldsymbol{y}}_{gk} \cdot \boldsymbol{g}_k}{\bar{\boldsymbol{y}}_{gk} \cdot (\boldsymbol{B}\bar{\boldsymbol{y}}_{gk})} \tag{3.4.8}$$

using Eq. (3.4.2) (the strict line search method). Furthermore, generate a sequence of iterates for $k \in \mathbb{N}$ using

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \bar{\epsilon}_{g\,k-1}\bar{\boldsymbol{y}}_{g\,k-1}, \tag{3.4.9}$$

$$\boldsymbol{g}_k = \boldsymbol{g}_{k-1} + \bar{\epsilon}_{g\,k-1}\boldsymbol{B}\bar{\boldsymbol{y}}_{g\,k-1}, \tag{3.4.10}$$

$$\beta_k = \frac{\boldsymbol{g}_k \cdot \boldsymbol{g}_k}{\boldsymbol{g}_{k-1} \cdot \boldsymbol{g}_{k-1}}, \tag{3.4.11}$$

$$\bar{\boldsymbol{y}}_{gk} = -\boldsymbol{g}_k + \beta_k\bar{\boldsymbol{y}}_{g\,k-1}. \tag{3.4.12}$$

In this case, $\bar{\boldsymbol{y}}_{g\,k-1}$ and $\bar{\boldsymbol{y}}_{gk}$ are conjugates (Practice **3.3**).

Equation (3.4.11) is called the Fletcher–Reeves formula. Moreover, its equivalent

$$\beta_k = \frac{\boldsymbol{g}_k \cdot (\boldsymbol{g}_k - \boldsymbol{g}_{k-1})}{\boldsymbol{g}_{k-1} \cdot \boldsymbol{g}_{k-1}}$$

is called the Polak–Ribière formula. Several formulae other than these are known. These formulae may be equivalent with respect to second-order optimization problems (Problem 3.4.2) but give different results when $\boldsymbol{g}_k = \boldsymbol{g}(\boldsymbol{x}_k)$ is used in non-linear optimization problems which are not second-order.

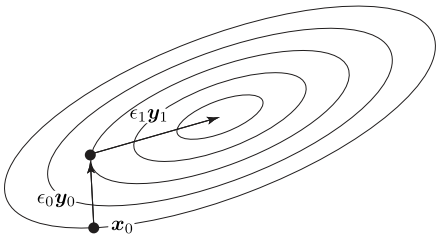


Fig. 3.9: Search vector obtained via conjugate gradient method.

3.5 Newton Method

In the gradient method, the gradient $\boldsymbol{g}$ was used to obtain the search direction. The step size was determined in order to satisfy the strict line search method, the Armijo criterion or the Wolfe criterion. In what follows, we shall consider a method for obtaining the search direction and step size simultaneously by using $\boldsymbol{g}$ and the Hesse matrix $\boldsymbol{H}$. This method is called the Newton method. This technique is used to obtain a search vector $\boldsymbol{y}_g \in X$ by ignoring $\boldsymbol{o}\left(\|\boldsymbol{y}_g\|_X\right)$ in the Taylor expansion of $\boldsymbol{g} \cdot \boldsymbol{y}$ with respect to an arbitrary $\boldsymbol{y} \in X$ about $\boldsymbol{x}_k$, i.e.,

$$\boldsymbol{g}\left(\boldsymbol{x}_k + \boldsymbol{y}_g\right) \cdot \boldsymbol{y} = \boldsymbol{g}\left(\boldsymbol{x}_k\right) \cdot \boldsymbol{y} + \boldsymbol{y}_g \cdot \left(\boldsymbol{H}\left(\boldsymbol{x}_k\right) \boldsymbol{y}\right) + \boldsymbol{o}\left(\|\boldsymbol{y}_g\|_X\right)$$

and then letting

$$\boldsymbol{g}\left(\boldsymbol{x}_k + \boldsymbol{y}_g\right) \cdot \boldsymbol{y} = \boldsymbol{g}\left(\boldsymbol{x}_k\right) \cdot \boldsymbol{y} + \boldsymbol{y}_g \cdot \left(\boldsymbol{H}\left(\boldsymbol{x}_k\right) \boldsymbol{y}\right) = 0.$$

In other words, the Newton method can be formally described as follows.

Problem 3.5.1 (Newton method) Let $X = \mathbb{R}^d$. Let the gradient and the Hesse matrix of f at $\boldsymbol{x}_k \in X$ which is not a local minimum point with respect to $f \in C^2\left(X; \mathbb{R}\right)$ be $\boldsymbol{g}\left(\boldsymbol{x}_k\right)$ and $\boldsymbol{H}\left(\boldsymbol{x}_k\right)$, respectively. In this case, obtain $\boldsymbol{y}_g \in X$ such that

$$\boldsymbol{y}_g \cdot \left(\boldsymbol{H}\left(\boldsymbol{x}_k\right) \boldsymbol{y}\right) = -\boldsymbol{g}\left(\boldsymbol{x}_k\right) \cdot \boldsymbol{y} \tag{3.5.1}$$

is satisfied for all $\boldsymbol{y} \in X$. □

The Newton method is the gradient method when the positive definite real symmetric matrix $\boldsymbol{A}$ used in the gradient method (Definition 3.3.1) is replaced by a Hesse matrix and such that $\bar{\epsilon}_g$ is taken to be unity. The following result can be obtained from the Newton method.

Theorem 3.5.2 (Newton method) Suppose f is twice differentiable in the neighborhood of the local minimum point $\boldsymbol{x}$ and the Hesse matrix $\boldsymbol{H}$ is Lipschitz continuous (Definition 4.3.1). Moreover, $\boldsymbol{H}\left(\boldsymbol{x}\right)$ is assumed to be positive definite. In this case, with a point sufficiently close to a local minimum point taken to be $\boldsymbol{x}_0$, the sequence of iterates $\left\{\boldsymbol{x}_k\right\}_{k \in \mathbb{N}}$ generated by the Newton method is second-order convergent. □

Proof Let the local minimum point be $\boldsymbol{x}$. From the fact that the Hesse matrix $\boldsymbol{H}$ is Lipschitz continuous around $\boldsymbol{x}$ and $\boldsymbol{H}(\boldsymbol{x})$ is regular, a point $\boldsymbol{x}_k$ sufficiently close to a local minimum point can be selected such that

$$\left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\left(\boldsymbol{H}(\boldsymbol{x}_k)-\boldsymbol{H}(\boldsymbol{x})\right)\right\|_{\mathbb{R}^{d\times d}} \leq \left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\right\|_{\mathbb{R}^{d\times d}} \beta \left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d} < \frac{1}{2} \tag{3.5.2}$$

is satisfied with respect to some $\beta > 0$, where

$$\left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\right\|_{\mathbb{R}^{d\times d}} = \max_{\boldsymbol{y}\in\mathbb{R}^d,\ \|\boldsymbol{y}\|_{\mathbb{R}^d}=1} \left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\,\boldsymbol{y}\right\|_{\mathbb{R}^d}.$$

In view of the above relationships, and using a standard result in Banach perturbation theory (cf. [**?**, p. 240]), we have that

$$\left\|\boldsymbol{H}^{-1}(\boldsymbol{x}_k)\right\|_{\mathbb{R}^{d\times d}} \leq \frac{\left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\right\|_{\mathbb{R}^{d\times d}}}{1-\left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\left(\boldsymbol{H}(\boldsymbol{x}_k)-\boldsymbol{H}(\boldsymbol{x})\right)\right\|_{\mathbb{R}^{d\times d}}} < 2\left\|\boldsymbol{H}^{-1}(\boldsymbol{x})\right\|_{\mathbb{R}^{d\times d}}. \tag{3.5.3}$$

Now, using $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$, Eq. (3.5.1) and $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}_{\mathbb{R}^d}$, we obtain

$$\begin{aligned}\boldsymbol{x}_{k+1}-\boldsymbol{x} &= \boldsymbol{x}_k - \boldsymbol{H}^{-1}(\boldsymbol{x}_k)\,\boldsymbol{g}(\boldsymbol{x}_k) - \boldsymbol{x} + \boldsymbol{H}^{-1}(\boldsymbol{x}_k)\,\boldsymbol{g}(\boldsymbol{x}) \\ &= -\boldsymbol{H}^{-1}(\boldsymbol{x}_k)\left\{\boldsymbol{g}(\boldsymbol{x}_k)-\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{H}(\boldsymbol{x}_k)(\boldsymbol{x}_k-\boldsymbol{x})\right\}.\end{aligned} \tag{3.5.4}$$

On the other hand, we have

$$\begin{aligned}&\left\|\boldsymbol{g}(\boldsymbol{x}_k)-\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{H}(\boldsymbol{x}_k)(\boldsymbol{x}_k-\boldsymbol{x})\right\|_{\mathbb{R}^d} \\ &= \left\|\int_0^1 \left(\boldsymbol{H}(\boldsymbol{x}+t(\boldsymbol{x}_k-\boldsymbol{x}))-\boldsymbol{H}(\boldsymbol{x}_k)\right)(\boldsymbol{x}_k-\boldsymbol{x})\,\mathrm{d}t\right\|_{\mathbb{R}^d} \\ &\leq \left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d}\int_0^1 \left\|\boldsymbol{H}(\boldsymbol{x}+t(\boldsymbol{x}_k-\boldsymbol{x}))-\boldsymbol{H}(\boldsymbol{x}_k)\right\|_{\mathbb{R}^{d\times d}}\,\mathrm{d}t \\ &\leq \left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d}\int_0^1 \beta\left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d}(1-t)\,\mathrm{d}t \\ &= \frac{1}{2}\beta\left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d}^2.\end{aligned}$$

Therefore, from Eq. (3.5.2), Eq. (3.5.3) and Eq. (3.5.4), it follows that

$$\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}\right\|_{\mathbb{R}^d} \leq \frac{1}{2}\left\|\boldsymbol{H}^{-1}(\boldsymbol{x}_k)\right\|_{\mathbb{R}^{d\times d}}\beta\left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d}^2 < \frac{1}{2}\left\|\boldsymbol{x}_k-\boldsymbol{x}\right\|_{\mathbb{R}^d}. \tag{3.5.5}$$

The relationship between the right-most side and the left-hand side of Eq. (3.5.5) confirms the convergence of the sequence of iterates to the local minimum point $\boldsymbol{x}$. Meanwhile, the quadratic convergence of the method follows from the relationship between the first right-hand side and the left-hand side of Eq. (3.5.5). □

An example of an algorithm using the Newton method is shown below. Figure 3.10 illustrates an overview of the method.

Algorithm 3.5.3 (Newton method) In Problem 3.1.1, f_0 is written as f and all inequality constraints are taken to be inactive.

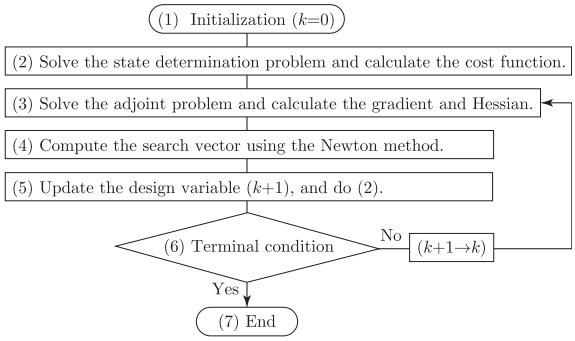


Fig. 3.10: Newton algorithm.

(1) Determine the initial value $\boldsymbol{x}_0$ and convergence criterion value ϵ_0. Set $k = 0$.

(2) Solve the state determination problem and calculate $f(\boldsymbol{x}_k)$.

(3) By solving the adjoint problem with respect to f, calculate $\boldsymbol{g}(\boldsymbol{x}_k)$ and $\boldsymbol{H}(\boldsymbol{x}_k)$.

(4) Calculate $\boldsymbol{y}_g$ using Eq. (3.5.1).

(5) Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$. Solve the state determination problem and calculate $f(\boldsymbol{x}_{k+1})$.

(6) Check the termination condition $|f_0(\boldsymbol{x}_{k+1}) - f_0(\boldsymbol{x}_k)| \leq \epsilon_0$.

- Proceed to (7) when the termination condition is satisfied.
- Otherwise, substitute $k+1$ into k and return to (3).

(7) Complete the calculation.

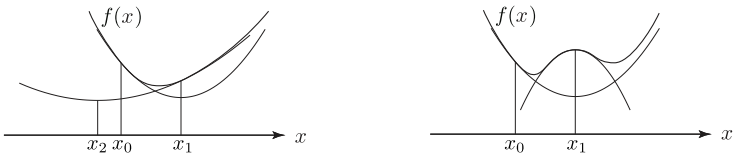
□

Let us emphasize the following properties of the Newton method.

Remark 3.5.4 (Newton method) The Newton method has the following quality:

(1) The Newton method requires the Hesse matrix. The amount of calculation of the Hesse matrix is proportional to the square of the design variable if the matrix is dense. However, when the Hesse matrix is a diagonal matrix, it is proportional to the design variable. Actually, the Hesse matrix in Problem 1.1.4 in Chap. 1 was a diagonal matrix.

(2) The Newton method has convergence of order two (Theorem 3.5.2).



(a) Non-convergent case. (b) Convergent case at a local maximizer.

Fig. 3.11: Newton method.

(3) If the Hesse matrix is not positive definite, there may be cases when convergence does not occur with the Newton method. Moreover, if it is indefinite, it may converge to the local maximum value (Fig. 3.11).

(4) When the Hesse matrix $\boldsymbol{H}$ is a singular (non-invertible) matrix, or there is a large condition number (rate of the maximum eigenvalue to the minimum eigenvalue) of the matrix even if it is not singular, the calculation of the inverse matrix becomes difficult. In fact, if the Hesse matrix is close to a singular matrix, the inverted matrix may be unstable and the solution may diverge.

□

Looking at it in this way, the gradient method (Problem 3.3.1) is a method which replaces the Hesse matrix in the Newton method (Problem 3.5.1) with a positive definite real symmetric matrix. Hence, by updating the positive definite symmetric matrices so that they are asymptotic to the Hesse matrix, gradient methods with qualities similar to the Newton method can be studied. These are called quasi-Newton methods. The following are among the known representative updated equations. For details, we refer the interested readers to textbooks on mathematical programming.

- Davidon–Fletcher–Powell method
- Broyden–Fletcher–Goldfarb–Shanno method
- Broyden method

The principle of the Newton method is also used when seeking for solutions of non-linear equations. In such a case, it is also called Newton–Raphson method. The Newton–Raphson method will also be used in Algorithm 3.7.6 which will be shown later. Hence, let us provide an explanation for it here. Problems to which the Newton–Raphson method applies are such as the one below.

Problem 3.5.5 (Non-linear equation) Let $X = \mathbb{R}^d$. With respect to $\boldsymbol{f} \in C^1\left(\mathbb{R}^d; \mathbb{R}^d\right)$, obtain $\boldsymbol{x} \in X$ such that the equation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}_{\mathbb{R}^d} \tag{3.5.6}$$

is satisfied. □

Let the trial point of Problem 3.5.5 be written as $\boldsymbol{x}_k$ for $k \in \mathbb{N}$. Also, assume that $\boldsymbol{f}$ evaluated at $\boldsymbol{x}_k$ and its gradient $\boldsymbol{G} = (\partial f_i/\partial x_j)_{(i,j)\in\{1,\ldots,d\}^2}$ are computable. Under these assumptions, find $\boldsymbol{y}_g$ such that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ becomes $\boldsymbol{x}$. The Taylor expansion of $\boldsymbol{f}$ about $\boldsymbol{x}_k$ can be written as

$$\boldsymbol{f}\left(\boldsymbol{x}_k + \boldsymbol{y}_g\right) = \boldsymbol{f}\left(\boldsymbol{x}_k\right) + \boldsymbol{G}\left(\boldsymbol{x}_k\right)\boldsymbol{y}_g + o\left(\left\|\boldsymbol{y}_g\right\|_{\mathbb{R}^d}\right).$$

Here, if $o\left(\left\|\boldsymbol{y}_g\right\|_{\mathbb{R}^d}\right)$ is ignored, then we will get

$$\boldsymbol{f}\left(\boldsymbol{x}_k + \boldsymbol{y}_g\right) = \boldsymbol{f}\left(\boldsymbol{x}_k\right) + \boldsymbol{G}\left(\boldsymbol{x}_k\right)\boldsymbol{y}_g = \boldsymbol{0}_{\mathbb{R}^d}. \tag{3.5.7}$$

Consider the following problem based on Eq. (3.5.7).

Problem 3.5.6 (Newton–Raphson method) Suppose the function value $\boldsymbol{f}\left(\boldsymbol{x}_k\right)$ and its gradient $\boldsymbol{G}\left(\boldsymbol{x}_k\right)$ at the trial point $\boldsymbol{x}_k$ are given with respect to Problem 3.5.5. Obtain the search vector using

$$\boldsymbol{y}_g = -\boldsymbol{G}^{-1}\left(\boldsymbol{x}_k\right)\boldsymbol{f}\left(\boldsymbol{x}_k\right). \tag{3.5.8}$$

□

The Newton–Raphson method is a method for obtaining a sequence of iterates $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ which converges to the solution $\boldsymbol{x}$ of Problem 3.5.5 from $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ using the solution $\boldsymbol{y}_g$ of Problem 3.5.6. If $\boldsymbol{g}$ and $\boldsymbol{H}$ of the Newton method (Problem 3.5.1) are replaced by $\boldsymbol{f}$ and $\boldsymbol{G}$, respectively, it should be understood that it agrees with the Newton–Raphson method (Problem 3.5.6).

Before we end this section, let us consider the Newton method in the case of using the second-order derivative of a cost function derived through the Lagrange multiplier method as explained in the last part of Section 1.1.6. Particularly, we assume that the second-order derivative of a cost function is obtained as the Hesse gradient in Eq. (1.1.49). In this case, $\boldsymbol{H}_0\boldsymbol{b}_1$ is obtained as $\boldsymbol{g}_{\mathrm{H}0}\left(\boldsymbol{a}, \boldsymbol{b}_1\right)$, then we can consider the gradient method using $\boldsymbol{g}_{\mathrm{H}0}\left(\boldsymbol{a}, \boldsymbol{b}_1\right)$ as the gradient. However, in order to obtain $\boldsymbol{g}_{\mathrm{H}0}\left(\boldsymbol{a}, \boldsymbol{b}_1\right)$, the adjoint problem (Problem 1.1.6) with respect to the first derivative of the cost function must be solved. Moreover, to solve the adjoint problem, $\boldsymbol{b}_1$ should be given. Considering these conditions, we proceed with solving the following problem.

Problem 3.5.7 (Newton method using Hesse gradient) Let $X = \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ and c_a be a positive definite real symmetric matrix and a positive constant. Moreover, let the gradient, the search vector and the Hesse gradient of f at $\boldsymbol{x}_k \in X$ which is not a local minimum point with respect to $f \in C^2\left(X; \mathbb{R}\right)$ be $\boldsymbol{g}\left(\boldsymbol{x}_k\right)$, $\bar{\boldsymbol{y}}_g$ and $\boldsymbol{g}_{\mathrm{H}}\left(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g\right)$, respectively. In this case, obtain $\boldsymbol{y}_g \in X$ such that

$$\boldsymbol{y}_g \cdot \left(c_a\boldsymbol{A}\left(\boldsymbol{x}_k\right)\boldsymbol{y}\right) = -\left(\boldsymbol{g}\left(\boldsymbol{x}_k\right) + \boldsymbol{g}_{\mathrm{H}}\left(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g\right)\right) \cdot \boldsymbol{y} \tag{3.5.9}$$

is satisfied for all $\boldsymbol{y} \in X$. □

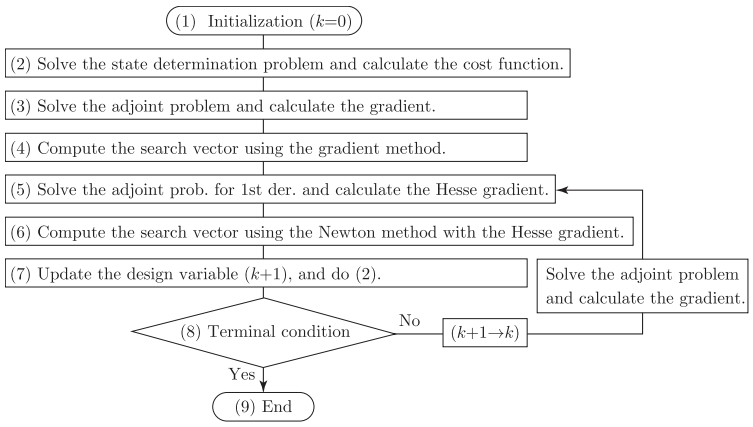


Fig. 3.12: Algorithm of Newton method using Hesse gradient.

The solution $\boldsymbol{y}_g$ of Problem 3.5.7 accords with the solution of the Newton method if $c_a \boldsymbol{A}(\boldsymbol{x}_k) = \boldsymbol{I}$ and $\bar{\boldsymbol{y}}_g = \boldsymbol{y}_g$. An example of an algorithm using the solution of Problem 3.5.7 is shown below. Figure 3.12 illustrates an overview of the method.

Algorithm 3.5.8 (Newton method using Hesse gradient)
In Problem 3.1.1, f_0 is written as f and all inequality constraints are taken to be inactive.

(1) Determine the initial value $\boldsymbol{x}_0$ and convergence criterion value ϵ_0. Set $k = 0$.

(2) Solve the state determination problem and calculate $f(\boldsymbol{x}_k)$.

(3) By solving the adjoint problem with respect to f, calculate $\boldsymbol{g}(\boldsymbol{x}_k)$.

(4) Calculate $\boldsymbol{y}_g$ using the gradient method (Eq. (3.3.6)).

(5) By solving the adjoint problem with respect to f', calculate the Hesse gradient $\boldsymbol{g}_{\mathrm{H}}(\boldsymbol{x}_k, \boldsymbol{y}_g)$.

(6) Calculate $\boldsymbol{y}_g$ by the Newton method (Eq. (3.5.9)) using $\boldsymbol{g}_{\mathrm{H}}(\boldsymbol{x}_k, \boldsymbol{y}_g)$.

(7) Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$. Solve the state determination problem and calculate $f(\boldsymbol{x}_{k+1})$.

(8) Check the termination condition $|f_0(\boldsymbol{x}_{k+1}) - f_0(\boldsymbol{x}_k)| \leq \epsilon_0$.

- Proceed to (9) when the termination condition is satisfied.
- Otherwise, substitute $k+1$ into k, solve the adjoint problem with respect to f, calculate $\boldsymbol{g}(\boldsymbol{x}_k)$ and return to (5).

(9) Complete the calculation.

□

3.6 Augmented Function Methods

Numerical solutions with respect to unconstrained optimization problems were considered from Sect. 3.3 to Sect. 3.5. Beyond this section, we shall revert to Problem 3.1.1 and consider the case when the inequality constraint becomes active at the local minimum point. We start this section by considering the method of replacing Problem 3.1.1 with an unconstrained problem by adding constraint functions multiplied by a constant representing weight to the objective function. Methods such as this are called augmented function methods. Methods for obtaining the minimum value of augmented functions make use of the solutions of unconstrained optimization problems shown from Sect. 3.3 to Sect. 3.5.

An augmented function method is a method that sets a point which satisfies all inequality constraints (inner point) as the initial point and utilizes an expansion function designed in a way that it would not fall outside of the admissible set. The method of obtaining a solution to Problem 3.1.1 by using the convergence point of the sequence of iterates $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ obtained in the following way is called the barrier method or inner point method.

Problem 3.6.1 (Barrier method, inner point method) Let $\{\rho_k\}_{k\in\mathbb{N}}$ be a positive monotonically decreasing sequence. Suppose that $\boldsymbol{x}_0 \in X$ is given such that the inequalities $f_1(\boldsymbol{x}_0) < 0, \ldots, f_m(\boldsymbol{x}_0) < 0$ hold. For $k \in \mathbb{N}$, provide a value for ρ_k and trial point $\boldsymbol{x}_{k-1}$ and obtain $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \boldsymbol{y}$ such that

$$\min_{\boldsymbol{y}\in X} \left\{ \hat{f}_k(\boldsymbol{x}_k, \rho_k) = f_0(\boldsymbol{x}_k) - \rho_k \sum_{i=1}^{m} \log\left(-f_i(\boldsymbol{x}_k)\right) \right\}.$$

□

There is another augmented function method which uses an initial point that does not satisfy the inequality constraint conditions and an expansion function such that the trial point is pushed toward being within the admissible set. The method of obtaining a solution for Problem 3.1.1 from the convergence point of the sequence of iterates $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ obtained in the following way is called the penalty method or outer point method.

Problem 3.6.2 (Penalty method, outer point method) Let $\{\rho_k\}_{k\in\mathbb{N}}$ be a positive monotonically increasing sequence. Suppose that $\boldsymbol{x}_0 \in X$ is given. For $k \in \mathbb{N}$, provide a value for ρ_k and trial point $\boldsymbol{x}_{k-1}$ and obtain $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \boldsymbol{y}$ such that

$$\min_{\boldsymbol{y}\in X} \left\{ \hat{f}_k(\boldsymbol{x}_k, \rho_k) = f_0(\boldsymbol{x}_k) + \rho_k \sum_{i\in\{1,\ldots,m\}} \max\left\{0, f_i(\boldsymbol{x}_k)\right\} \right\}.$$

□

From the definitions given above, the augmented function method should be easy to use from the fact that the principles are clear. However, in order to use this method, it is necessary to choose an appropriate monotonically decreasing sequence or monotonically increasing sequence $\{\rho_k\}_{k\in\mathbb{N}}$ depending on the problem. In this book we will not touch upon the details of its selection method.

3.7 Gradient Method for Constrained Problems

In this section, and in Sect. 3.8, we shall turn our attention to a method that employs the KKT conditions. The algorithms that will be shown here will be used in Chap. 7 and beyond. To begin with, let us consider the gradient method for constrained problems.

The admissible set for which the inequality constraints are satisfied with respect to Problem 3.1.1 is written as

$$S = \{\boldsymbol{x} \in X \mid f_1(\boldsymbol{x}) \leq 0, \ldots, f_m(\boldsymbol{x}) \leq 0\}. \tag{3.7.1}$$

Moreover, for each $\boldsymbol{x} \in S$, we shall denote by

$$I_{\mathrm{A}}(\boldsymbol{x}) = \{i \in \{1, \ldots, m\} \mid f_i(\boldsymbol{x}) \geq 0\} = \left\{i_1, \ldots, i_{|I_{\mathrm{A}}(\boldsymbol{x})|}\right\} \tag{3.7.2}$$

the set of subscripts for active constraints. When there is no confusion, $I_{\mathrm{A}}(\boldsymbol{x}_k)$ is written as I_{A}.

The gradient method was considered as a Newton method when cost function f is approximated by a second-order approximate function $q(\boldsymbol{y})$ in Eq. (3.3.8) around the trial point $\boldsymbol{x}_k \in S$, for $k \in \mathbb{N}$. So, with respect to Problem 3.1.1, assume the cost function to be $q(\boldsymbol{y})$ in Eq. (3.3.8) and consider the problem in which the inequality constraint is approximated by a first-order function using the gradient such as the following.

Problem 3.7.1 (Gradient method for constrained problems) For a trial point $\boldsymbol{x}_k \in S$ in Problem 3.1.1, let $f_0(\boldsymbol{x}_k)$, $f_{i_1}(\boldsymbol{x}_k) = 0, \ldots, f_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k) = 0$, and $\boldsymbol{g}_0(\boldsymbol{x}_k)$, $\boldsymbol{g}_{i_1}(\boldsymbol{x}_k), \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k)$ be given. Moreover, let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ be a positive definite real symmetric matrix and c_a be a positive constant. Obtain $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ which satisfies

$$q(\boldsymbol{y}_g) = \min_{\boldsymbol{y}\in X}\Big\{q(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}\cdot(c_a\boldsymbol{A}\boldsymbol{y}) + \boldsymbol{g}_0(\boldsymbol{x}_k)\cdot\boldsymbol{y} + f_0(\boldsymbol{x}_k) \;\Big|\; f_i(\boldsymbol{x}_k) + \boldsymbol{g}_i(\boldsymbol{x}_k)\cdot\boldsymbol{y} \leq 0 \text{ for } i \in I_{\mathrm{A}}(\boldsymbol{x}_k)\Big\}.$$

□

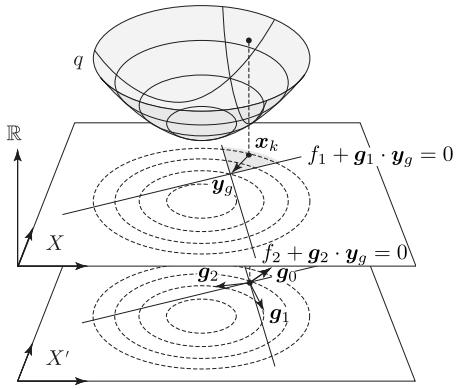


Fig. 3.13: Gradient method for constrained problems.

Problem 3.7.1 can be classified as a second-order optimization problem based on the classification of optimization problems in Section 2.2. Furthermore, with the fact that $\boldsymbol{A}$ is a positive definite symmetric real matrix, it is a convex optimization problem. Therefore $\boldsymbol{y}_g$, which satisfies the KKT conditions with respect to this problem, is the minimum point of Problem 3.7.1 (Fig. 3.13). Let us examine a method for finding $\boldsymbol{y}_g$.

Let the Lagrange function of Problem 3.7.1 be

$$\mathscr{L}_{\mathrm{Q}}\left(\boldsymbol{y}, \boldsymbol{\lambda}_{k+1}\right) = q\left(\boldsymbol{y}\right) + \sum_{i \in I_{\mathrm{A}}\left(\boldsymbol{x}_k\right)} \lambda_{i\,k+1}\left(f_i\left(\boldsymbol{x}_k\right) + \boldsymbol{g}_i\left(\boldsymbol{x}_k\right) \cdot \boldsymbol{y}\right). \tag{3.7.3}$$

The KKT conditions for Problem 3.7.1's minimum point $\boldsymbol{y}_g$ are as follows:

$$c_a \boldsymbol{A}\boldsymbol{y}_g + \boldsymbol{g}_0\left(\boldsymbol{x}_k\right) + \sum_{i \in I_{\mathrm{A}}\left(\boldsymbol{x}_k\right)} \lambda_{i\,k+1}\boldsymbol{g}_i\left(\boldsymbol{x}_k\right) = \boldsymbol{0}_{X'}, \tag{3.7.4}$$

$$f_i\left(\boldsymbol{x}_k\right) + \boldsymbol{g}_i\left(\boldsymbol{x}_k\right) \cdot \boldsymbol{y}_g \leq 0 \quad \text{for } i \in I_{\mathrm{A}}\left(\boldsymbol{x}_k\right), \tag{3.7.5}$$

$$\lambda_{i\,k+1}\left(f_i\left(\boldsymbol{x}_k\right) + \boldsymbol{g}_i\left(\boldsymbol{x}_k\right) \cdot \boldsymbol{y}_g\right) = 0 \quad \text{for } i \in I_{\mathrm{A}}\left(\boldsymbol{x}_k\right), \tag{3.7.6}$$

$$\lambda_{i\,k+1} \geq 0 \quad \text{for } i \in I_{\mathrm{A}}\left(\boldsymbol{x}_k\right). \tag{3.7.7}$$

If the inequality constraints are assumed to be active with respect to $i \in I_{\mathrm{A}}\left(\boldsymbol{x}_k\right)$, Eq. (3.7.4) and Eq. (3.7.5) become

$$\begin{pmatrix} c_a \boldsymbol{A} & \boldsymbol{G}^{\top} \\ \boldsymbol{G} & \boldsymbol{0}_{\mathbb{R}^{|I_{\mathrm{A}}| \times |I_{\mathrm{A}}|}} \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_g \\ \boldsymbol{\lambda}_{k+1} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{g}_0 \\ \left(f_i\right)_{i \in I_{\mathrm{A}}} \end{pmatrix}, \tag{3.7.8}$$

where

$$\boldsymbol{G}^{\top} = \begin{pmatrix} \boldsymbol{g}_{i_1}\left(\boldsymbol{x}_k\right) & \cdots & \boldsymbol{g}_{i_{|I_{\mathrm{A}}\left(\boldsymbol{x}_k\right)|}}\left(\boldsymbol{x}_k\right) \end{pmatrix}.$$

In this case if $\boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$ are linearly independent, Eq. (3.7.8) is solvable about $\left(\boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}\right)$. With respect to the solutions of these simultaneous linear

equations, denote by

$$I_{\mathrm{I}}(\boldsymbol{x}_k) = \{\, i \in I_{\mathrm{A}}(\boldsymbol{x}_k) \mid \lambda_{i\,k+1} < 0 \} \tag{3.7.9}$$

the set of inactive constraint conditions. If it happens that $I_{\mathrm{I}}(\boldsymbol{x}_k) \neq \emptyset$, $I_{\mathrm{A}}(\boldsymbol{x}_k) \setminus I_{\mathrm{I}}(\boldsymbol{x}_k)$ is replaced with $I_{\mathrm{A}}(\boldsymbol{x}_k)$ and Eq. (3.7.8) should be solved again. The pair $(\boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}) \in X \times \mathbb{R}^{|I_{\mathrm{A}}|}$ obtained in this way satisfies Eq. (3.7.4) to Eq. (3.7.7). The iterative method, leaving only the active constraints for each iteration, is called the active set method [**?**, Section 10.10.6, p. 447].

On the other hand, a method can be considered in which, instead of solving Eq. (3.7.8) directly, results in using the gradient method with respect to f_i for each $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$. This method is described as follows. The functions $\boldsymbol{g}_0$, $\boldsymbol{g}_{i_1}$, $\ldots$, $\boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$ are used and the gradient method is applied individually. In other words, $\boldsymbol{y}_{g0}$, $\boldsymbol{y}_{gi_1}$, $\ldots$, $\boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}}$ is sought so that

$$\boldsymbol{y}_{gi} = -(c_a \boldsymbol{A})^{-1} \boldsymbol{g}_i \tag{3.7.10}$$

is satisfied. Here, the Lagrange multiplier $\boldsymbol{\lambda}_{k+1} \in \mathbb{R}^{|I_{\mathrm{A}}|}$ is taken to be an unknown variable and

$$\boldsymbol{y}_g = \boldsymbol{y}_g(\boldsymbol{\lambda}_{k+1}) = \boldsymbol{y}_{g0} + \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{i\,k+1}\, \boldsymbol{y}_{gi}. \tag{3.7.11}$$

It can be verified that $\boldsymbol{y}_g$ satisfies the first row of Eq. (3.7.8). On the other hand, the second row of Eq. (3.7.8) becomes

$$\begin{aligned}
&\begin{pmatrix} \boldsymbol{g}_{i_1} \cdot \boldsymbol{y}_{gi_1} & \cdots & \boldsymbol{g}_{i_1} \cdot \boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}} \cdot \boldsymbol{y}_{gi_1} & \cdots & \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}} \cdot \boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}} \end{pmatrix} \begin{pmatrix} \lambda_{i_1\,k+1} \\ \vdots \\ \lambda_{i_{|I_{\mathrm{A}}|}\,k+1} \end{pmatrix} \\
&\quad = -\begin{pmatrix} f_{i_1} + \boldsymbol{g}_{i_1} \cdot \boldsymbol{y}_{g0} \\ \vdots \\ f_{i_{|I_{\mathrm{A}}|}} + \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}} \cdot \boldsymbol{y}_{g0} \end{pmatrix},
\end{aligned}$$

which can equivalently be written as

$$\left(\boldsymbol{g}_i \cdot \boldsymbol{y}_{gj}\right)_{(i,j) \in I_{\mathrm{A}}^2} \left(\lambda_{j\,k+1}\right)_{j \in I_{\mathrm{A}}} = -\left(f_i + \boldsymbol{g}_i \cdot \boldsymbol{y}_{g0}\right)_{i \in I_{\mathrm{A}}}. \tag{3.7.12}$$

Again, in this case, if $\boldsymbol{g}_{i_1}$, $\ldots$, $\boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$ are linearly independent, then $\boldsymbol{\lambda}_{k+1}$ are uniquely determined by Eq. (3.7.12). The active constraint method is then applied with respect to the solution of these simultaneous linear equations. In other words, when the set $I_{\mathrm{I}}(\boldsymbol{x}_k)$ in Eq. (3.7.9) is non-empty, $I_{\mathrm{A}}(\boldsymbol{x}_k) \setminus I_{\mathrm{I}}(\boldsymbol{x}_k)$ is replaced by $I_{\mathrm{A}}(\boldsymbol{x}_k)$ and Eq. (3.7.8) is solved again. The pair $(\boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}) \in X \times \mathbb{R}^{|I_{\mathrm{A}}|}$ obtained in this way should satisfy Eq. (3.7.4) to Eq. (3.7.7).

Moreover, in Eq. (3.7.12), if all the values of active constraint functions f_{i_1}, $\ldots$, $f_{i_{|I_{\mathrm{A}}|}}$ are zero, even if all of $\boldsymbol{y}_{g0}$, $\boldsymbol{y}_{gi_1}$, $\ldots$, $\boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}}$ are multiplied by an

arbitrary constant, $\boldsymbol{\lambda}_{k+1}$ remains unchanged. This shows that even if the step size $\|\boldsymbol{y}_g\|_X$ has not been appropriately selected, $\boldsymbol{\lambda}_{k+1}$ can be obtained. This relationship is used in Sect. 3.7.2 in order to determine c_a such that the initial value of the step size becomes the desired size.

The above discussion is summarized as follows: the gradient method for constrained problems is an iterative method, whereby $\boldsymbol{y}_g$ is updated by either directly solving for the search vector $\boldsymbol{y}_g$ and the Lagrange multiplier $\boldsymbol{\lambda}_{k+1}$ using Eq. (3.7.8), or by solving for $\boldsymbol{y}_{gi}$ with Eq. (3.7.10) for each $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$, using those to obtain $\boldsymbol{\lambda}_{k+1}$ from Eq. (3.7.12), and substituting them into Eq. (3.7.11) in order to obtain $\boldsymbol{y}_g$.

Before showing specific algorithms, let us consider several situations. One is a situation whereby the inequality constraint of Problem 3.1.1 is replaced by equality constraint $f_i(\boldsymbol{x}) = 0$. In reality, the inequality constraints are treated in the same manner when the equality constraints are active, and so, this situation can always arise. This type of equality constraint can be replaced by two inequality constraints $f_i(\boldsymbol{x}) \leq 0$ and $-f_i(\boldsymbol{x}) \leq 0$. However, when these two inequality constraints are non-linear, determining $\boldsymbol{x}$ so that they are strictly satisfied is generally difficult. Hence, there is a need to determine a positive constant ϵ_i and relax the constraint such as by $|f_i(\boldsymbol{x})| \leq \epsilon_i$. In algorithms that will be shown later, only inequality constraints are assumed; it may be thought that there is no need to relax the constraints using ϵ_i. However, if inequality constraints are active, they have the same meaning as the equality constraints $f_i(\boldsymbol{x}) = 0$ and there is a need to relax the constraint using a positive constant ϵ_i.

Additionally, we suppose a situation in which all inequality constraints are satisfied at the initial point $\boldsymbol{x}_0$. If this type of condition is not satisfied, $\boldsymbol{x}_0 \in S$ which satisfies all the inequality constraints can be found by carrying out the following steps for pre-processing. If they cannot be found, there is a need to review the problem set-up.

(0) Let the cost function f_0 be zero and $\boldsymbol{g}_0$ be equal to the zero vector $\mathbf{0}_{\mathbb{R}^d}$ and then repeat the established steps in the algorithm that will be shown later until all the inequality constraints are satisfied.

3.7.1 Simple Algorithm

With all the things looked at already in mind, let us now examine a simple algorithm in the succeding discussion. In this section, the parameter c_a for adjusting the step size is given in advance, and an example of an algorithm when inequality constraint checks are not carried out after updating the design variables is shown. Figure 3.14 shows the flow diagram for the algorithm.

Algorithm 3.7.2 (Gradient method without parameter adjustment)
Obtain the local minimum point of Problem 3.1.1 in the following way:

(1) Determine the initial point $\boldsymbol{x}_0$ so that the inequality constraints $f_1(\boldsymbol{x}_0) \leq 0, \ldots, f_m(\boldsymbol{x}_0) \leq 0$ are satisfied. Determine the positive definite matrix

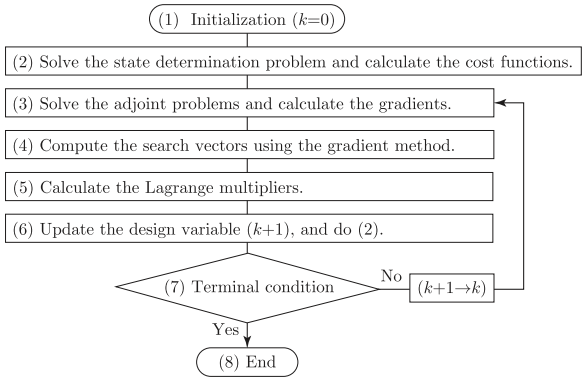


Fig. 3.14: Algorithm of gradient method for constraint problems without parameter adjustment.

$\boldsymbol{A}$ of Eq. (3.7.10), positive constant c_a for adjusting the step size, positive constant ϵ_0 used for the check of convergence of f_0 as well as the positive constants $\epsilon_1, \ldots, \epsilon_m$ providing the permissible ranges of $f_1, \ldots, f_m$. Set $k = 0$.

(2) Solve the state determination problem for $\boldsymbol{x}_k$ and calculate $f_0(\boldsymbol{x}_k)$, $f_1(\boldsymbol{x}_k), \ldots, f_m(\boldsymbol{x}_k)$. Moreover, let

$$I_{\mathrm{A}}(\boldsymbol{x}_k) = \{\, i \in \{1, \ldots, m\} \mid f_i(\boldsymbol{x}_k) \geq -\epsilon_i \,\}.$$

(3) Solve the adjoint problem with respect to $f_0, f_{i_1}, \ldots, f_{i_{|I_{\mathrm{A}}|}}$ and for $\boldsymbol{x}_k$, calculate $\boldsymbol{g}_0, \boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$.

(4) Calculate $\boldsymbol{y}_{g0}, \boldsymbol{y}_{gi_1}, \ldots, \boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}}$ with Eq. (3.7.10).

(5) Use Eq. (3.7.12) to seek $\boldsymbol{\lambda}_{k+1}$. If $I_{\mathrm{I}}(\boldsymbol{x}_k)$ in Eq. (3.7.9) is non-empty, replace $I_{\mathrm{A}}(\boldsymbol{x}_k) \setminus I_{\mathrm{I}}(\boldsymbol{x}_k)$ by $I_{\mathrm{A}}(\boldsymbol{x}_k)$ and solve Eq. (3.7.12) again.

(6) Use Eq. (3.7.11) to seek $\boldsymbol{y}_g$, and letting $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$, calculate $f_0(\boldsymbol{x}_{k+1})$, $f_1(\boldsymbol{x}_{k+1}), \ldots, f_m(\boldsymbol{x}_{k+1})$. Moreover, define

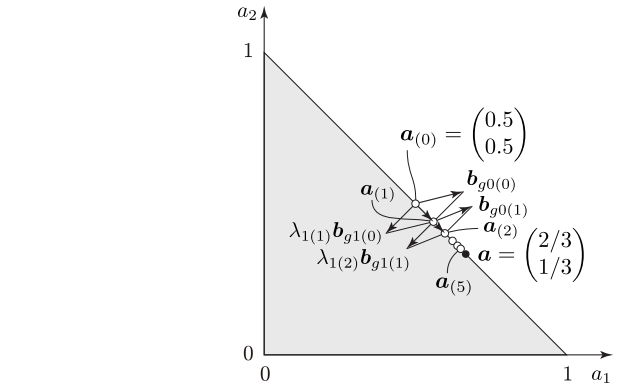
$$I_{\mathrm{A}}(\boldsymbol{x}_{k+1}) = \{\, i \in \{1, \ldots, m\} \mid f_i(\boldsymbol{x}_{k+1}) \geq -\epsilon_i \,\}.$$

(7) Check the terminal condition $|f_0(\boldsymbol{x}_{k+1}) - f_0(\boldsymbol{x}_k)| \leq \epsilon_0$.

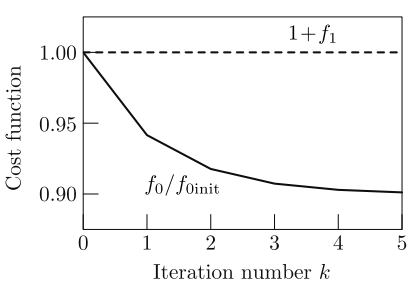
- Proceed to (8) when the terminal condition is satisfied.
- Otherwise substitute $k + 1$ into k and revert to (3).

(8) End the calculation.

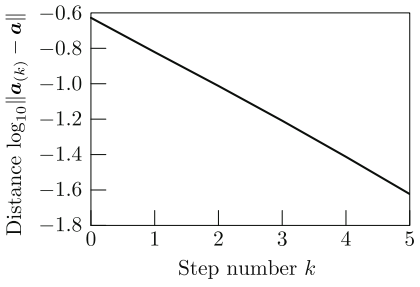
□



(a) Movement of trial point.



(b) History of cost function.



(c) Distance $\|\boldsymbol{a}_k - \boldsymbol{a}\|_{\mathbb{R}^2}$.

Fig. 3.15: Numerical example of mean compliance minimization problem via gradient method with respect to constraint problem without parameter adjustment.

Let us seek the trial points with respect to Exercise 1.1.7 (numerical example of mean compliance minimization problem) in Chap. 1 using Algorithm 3.7.2.

Exercise 3.7.3 (Mean compliance minimization problem)
Consider Exercise 1.1.7. Let the initial point be $\boldsymbol{a}_{(0)} = (1/2, 1/2)^\top$ and use Algorithm 3.7.2 in order to obtain the trial points for $k \in \{0, 1\}$. Here, the required matrix and numerical values should be determined appropriately. □

Answer The mean compliance $\tilde{f}_0(\boldsymbol{a})$ and volume constraint function $f_1(\boldsymbol{a})$ are given by

$$\tilde{f}_0(\boldsymbol{a}) = \frac{4}{a_1} + \frac{1}{a_2}, \tag{3.7.13}$$

$$f_1(\boldsymbol{a}) = a_1 + a_2 - 1 \tag{3.7.14}$$

respectively. Moreover, their cross-sectional derivative will be

$$\boldsymbol{g}_0 = -\begin{pmatrix} \frac{4}{a_1^2} \\ \frac{1}{a_2^2} \end{pmatrix}, \tag{3.7.15}$$

$$\boldsymbol{g}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{3.7.16}$$

Numerical values are sought along with Algorithm 3.7.2. Here, the design variable is written as $\boldsymbol{a}_{(k)}$ for each step number $k \in \mathbb{N}$. The same is true for $\boldsymbol{b}_{g0(k)}$, $\boldsymbol{b}_{g1(k)}$ and $\lambda_{1(k)}$.

(1) At initial point $\boldsymbol{a}_{(0)} = (1/2, 1/2)^\top$, $f_1\left(\boldsymbol{a}_{(0)}\right) = 0$ is satisfied. Let the positive definite matrix of Eq. (3.7.10) be $\boldsymbol{A} = \boldsymbol{I}$, and the positive constant for adjusting the step size be $c_a = 100$ (step size is $\left\|\boldsymbol{b}_{g(0)}\right\|_{\mathbb{R}^2} = 0.0848528$ from calculation shown later on) and $\epsilon_0 = 10^{-3}\tilde{f}_0\left(\boldsymbol{a}_{(0)}\right)$, $\epsilon_1 = 10^{-3}$. Set $k = 0$.

(2) Equations (3.7.13) and (3.7.14) give $\tilde{f}_0\left(\boldsymbol{a}_{(0)}\right) = 10$ and $f_1\left(\boldsymbol{a}_{(0)}\right) = 0$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(0)}\right) = \{1\}$.

(3) Equations (3.7.15) and (3.7.16) give $\boldsymbol{g}_{0(0)} = -(16, 4)^\top$, $\boldsymbol{g}_{1(0)} = (1, 1)^\top$.

(4) Equation (3.7.10 gives $\boldsymbol{b}_{g0(0)} = (0.16, 0.04)^\top$, $\boldsymbol{b}_{g1(0)} = -(0.01, 0.01)^\top$.

(5) Equation (3.7.12 gives $\boldsymbol{\lambda}_{1(1)} = 10$.

(6) Equation (3.7.11 gives $\boldsymbol{b}_{g(0)} = (0.06, -0.06)^\top$ and letting $\boldsymbol{a}_{(1)} = \boldsymbol{a}_{(0)} + \boldsymbol{b}_{g(0)} = (0.56, 0.44)^\top$ give $\tilde{f}_0\left(\boldsymbol{a}_{(1)}\right) = 9.41558$, $f_1\left(\boldsymbol{a}_{(1)}\right) = 0$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(1)}\right) = \{1\}$.

(7) $\left|\tilde{f}_0\left(\boldsymbol{a}_{(1)}\right) - \tilde{f}_0\left(\boldsymbol{a}_{(0)}\right)\right| = 0.584416 \geq \epsilon_0 = 0.01$ suggests that the terminal condition is not satisfied and hence substitute 1 into k and revert to (3).

(3) Equations (3.7.15) and (3.7.16) give $\boldsymbol{g}_{0(1)} = -(12.7551, 5.16529)^\top$, $\boldsymbol{g}_{1(1)} = (1, 1)^\top$.

(4) Equation (3.7.10 gives $\boldsymbol{b}_{g0(1)} = (0.127551, 0.0516529)^\top$ and $\boldsymbol{b}_{g1(1)} = -(0.01, 0.01)^\top$.

(5) Equation (3.7.12 gives $\boldsymbol{\lambda}_{1(2)} = 8.9602$.

(6) Equation (3.7.11 gives $\boldsymbol{b}_{g(1)} = (0.0379491, -0.0379491)^\top$ and letting $\boldsymbol{a}_{(2)} = \boldsymbol{a}_{(1)} + \boldsymbol{b}_{g(1)} = (0.597949, 0.402051)^\top$ gives $\tilde{f}_0\left(\boldsymbol{a}_{(2)}\right) = 9.17678$, $f_1\left(\boldsymbol{a}_{(2)}\right) = 0$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(1)}\right) = \{1\}$.

(7) $\left|\tilde{f}_0\left(\boldsymbol{a}_{(2)}\right) - \tilde{f}_0\left(\boldsymbol{a}_{(1)}\right)\right| = 0.238804 \geq \epsilon_0 = 0.01$ shows that the terminal condition is not satisfied and so substitute 2 into k and return to (3).

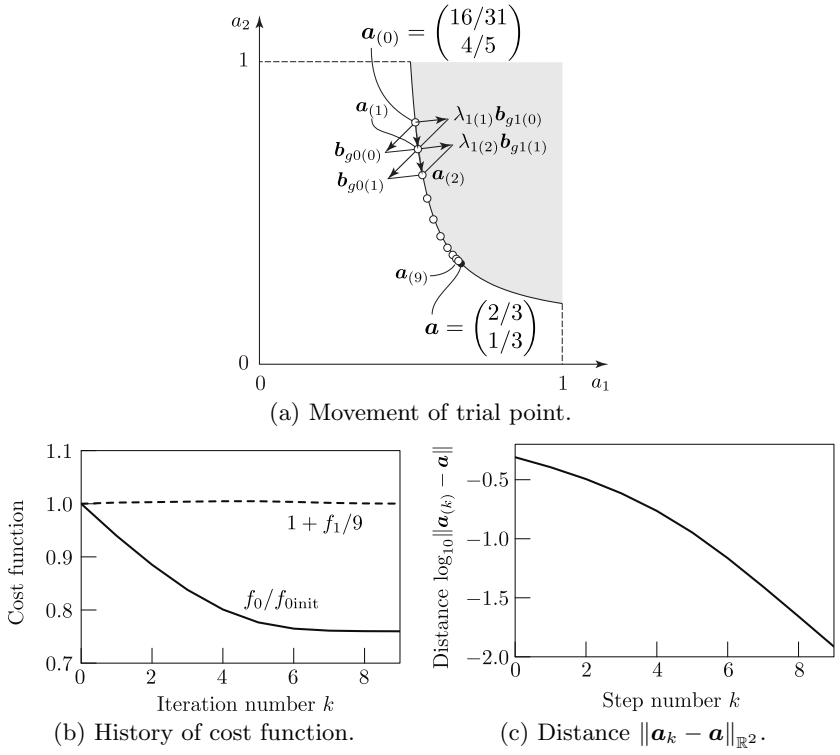
Figure 3.15 illustrates the above computations as well as the later computations. Here, $f_{0\,\mathrm{init}}$ denotes the value of f_0 at $k = 0$. □

Next, let us change the cost function and constraint function of Exercise 1.1.7.

Exercise 3.7.4 (Volume minimization problem) Let the cost function and constraint function be

$$f_0(\boldsymbol{a}) = a_1 + a_2, \tag{3.7.17}$$

$$\tilde{f}_1(\boldsymbol{a}) = \frac{4}{a_1} + \frac{1}{a_2} - 9, \tag{3.7.18}$$



(a) Movement of trial point.

(b) History of cost function.

(c) Distance $\|\boldsymbol{a}_k - \boldsymbol{a}\|_{\mathbb{R}^2}$.

Fig. 3.16: Numerical example of volume minimizing problem using gradient method without parameter adjustment.

respectively. In this case, under the constraint which satisfies $\tilde{f}_1(\boldsymbol{a}) \leq 0$, let the initial point with respect to the problem minimizing $f_0(\boldsymbol{a})$ (volume minimizing problem with mean compliance constraint) be $\boldsymbol{a}_{(0)} = (16/31, 4/5)^\top \approx (0.516, 0.8)^\top$ and use Algorithm 3.7.2 in order to obtain the trial point for $k \in \{0, 1\}$. Here, the required matrices and numerical values should be appropriately determined. □

Answer The cross-sectional derivatives of the cost functions $f_0(\boldsymbol{a})$ and $\tilde{f}_1(\boldsymbol{a})$ are

$$\boldsymbol{g}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \tag{3.7.19}$$

$$\boldsymbol{g}_1 = -\begin{pmatrix} \frac{4}{a_1^2} \\ \frac{1}{a_2^2} \end{pmatrix}. \tag{3.7.20}$$

The numerical values are sought alongside Algorithm 3.7.2. In this case, the design variables are again denoted by $\boldsymbol{a}_{(k)}$ for each step number $k \in \mathbb{N}$. The same applies to $\boldsymbol{b}_{g0(k)}$, $\boldsymbol{b}_{g1(k)}$ and $\lambda_{1(k)}$.

(1) $\tilde{f}_1\left(\boldsymbol{a}_{(0)}\right) = 0$ is satisfied when the initial point is $\boldsymbol{a}_{(0)} = (16/31, 4/5)^\top$. Let the positive definite matrix of Eq. (3.7.10) be $\boldsymbol{A} = \boldsymbol{I}$, positive constant $c_a = 10$ which adjusts the step size to be $c_a = 10$ (step size is $\left\|\boldsymbol{b}_{g(0)}\right\|_{\mathbb{R}^2} = 0.089113$ based on calculations shown later) and $\epsilon_0 = 10^{-3} f_0\left(\boldsymbol{a}_{(0)}\right)$, $\epsilon_1 = 9 \times 10^{-3}$. Moreover, let $k = 0$.

(2) Equations (3.7.17) and (3.7.18) give $f_0\left(\boldsymbol{a}_{(0)}\right) = 1.31613$ and $\tilde{f}_1\left(\boldsymbol{a}_{(0)}\right) = 0$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(0)}\right) = \{1\}$.

(3) Equations (3.7.15) and (3.7.16) give $\boldsymbol{g}_{0(0)} = (1,1)^\top$, $\boldsymbol{g}_{1(0)} = -(15.0156, 1.5625)^\top$.

(4) Equation (3.7.10) gives $\boldsymbol{b}_{g0(0)} = -(0.1, 0.1)^\top$ and $\boldsymbol{b}_{g1(0)} = (1.50156, 0.15625)^\top$.

(5) Equation (3.7.12) gives $\boldsymbol{\lambda}_{1(1)} = 0.0727397$.

(6) Equation (3.7.11) gives $\boldsymbol{b}_{g(0)} = (0.00922315, -0.0886344)^\top$ and lets $\boldsymbol{a}_{(1)} = \boldsymbol{a}_{(0)} + \boldsymbol{b}_{g(0)} = (0.525352, 0.711366)^\top$ gives $f_0\left(\boldsymbol{a}_{(1)}\right) = 1.23672$, $\tilde{f}_1\left(\boldsymbol{a}_{(1)}\right) = 0.019687$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(1)}\right) = \{1\}$.

(7) $\left|f_0\left(\boldsymbol{a}_{(1)}\right) - f_0\left(\boldsymbol{a}_{(0)}\right)\right| = 0.0794113 \geq \epsilon_0 = 0.00131613$ suggests that the terminal condition is not yet satisfied, so substitute 1 into k and revert to (3).

(3) Equations (3.7.15) and (3.7.16) give $\boldsymbol{g}_{0(1)} = (1,1)^\top$, $\boldsymbol{g}_{1(1)} = -(14.493, 1.97612)^\top$.

(4) Equation (3.7.10) gives $\boldsymbol{b}_{g0(1)} = -(0.1, 0.1)^\top$ and $\boldsymbol{b}_{g1(1)} = (1.4493, 0.197612)^\top$.

(5) Equation (3.7.12) gives $\boldsymbol{\lambda}_{1(2)} = 0.0778958$.

(6) Equation (3.7.11) gives $\boldsymbol{b}_{g(1)} = (0.0128945, -0.0846068)^\top$ and by letting $\boldsymbol{a}_{(2)} = \boldsymbol{a}_{(1)} + \boldsymbol{b}_{g(1)} = (0.538247, 0.626759)^\top$, $f_0\left(\boldsymbol{a}_{(2)}\right) = 1.16501$ and $\tilde{f}_1\left(\boldsymbol{a}_{(2)}\right) = 0.0270467$ can be obtained. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(1)}\right) = \{1\}$.

(7) From $\left|f_0\left(\boldsymbol{a}_{(2)}\right) - f_0\left(\boldsymbol{a}_{(1)}\right)\right| = 0.0717123 \geq \epsilon_0 = 0.00131613$, the terminal condition is seen to not be satisfied, 2 is substituted in for k and reverts to (3).

Figure 3.16 shows these results and later computed values. □

In Exercise 3.7.3, the constraints were always satisfied as $f_1\left(\boldsymbol{a}_{(1)}\right) = f_1\left(\boldsymbol{a}_{(2)}\right) = 0$. However, in Exercise 3.7.4, the constraints were not satisfied since $\tilde{f}_1\left(\boldsymbol{a}_{(1)}\right) = 0.019687$ and $\tilde{f}_1\left(\boldsymbol{a}_{(2)}\right) = 0.0459682$ and their excess values increased with each reiteration. Methods for preventing such a situation will be considered in the next section.

3.7.2 Complicated Algorithm

Let us consider adding the following type of function to a simple algorithm (Algorithm 3.7.2):

(i) A function for determining c_a such that, given the initial step size ϵ_g (or the objective function reduce rate α), $\left\|\boldsymbol{y}_g\right\|_X = \epsilon_g$.

(ii) A function for correcting $\boldsymbol{\lambda}_{k+1} = (\lambda_{i\,k+1})_{i \in I_{\mathrm{A}}(\boldsymbol{x}_{k+1})}$ such that, when design variable is updated to $\boldsymbol{x}_{k+1}$, $|f_i(\boldsymbol{x}_{k+1})| \leq \epsilon_i$ and $\lambda_{i\,k+1} \geq 0$ are satisfied with respect to $i \in I_{\mathrm{A}}(\boldsymbol{x}_{k+1})$.

(iii) A function for adjusting the permissible values $\epsilon_1, \ldots, \epsilon_m$ for constraint functions $f_1, \ldots, f_m$ with respect to the convergence check value ϵ_0 for cost function f_0.

(iv) A function for adjusting the step size $\|\boldsymbol{y}_g\|_X$ so that global convergence is guaranteed.

(i) above can be solved, as with Algorithm 3.3.5, by seeking c_a with Eq. (3.3.7). It is included in step (6) in Algorithm 3.7.6 which will be shown later.

Moreover, the following types of methods can be thought of with respect to (ii). Even if the value of $\boldsymbol{\lambda}_{k+1}$ calculated in step (5) of Algorithm 3.7.2 satisfies the KKT conditions of the gradient method for constrained problems (Problem 3.7.1), the non-linearity of active inequality constraint functions suggest that it is not necessarily satisfied in the specified permissible range at $\boldsymbol{x}_{k+1}$.

In order for it to be satisfied within the permissible range, where each of the inequality constraint is specified by $\boldsymbol{x}_{k+1}$, $\boldsymbol{\lambda}_{k+1}$ needs to be amended and this requires modifying Eq. (3.7.11), so that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ will be updated. In this case, let us consider setting $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_{k+1\,0}$, we provide $\boldsymbol{\lambda}_{k+1\,l}$ for $l \in \{0, 1, 2, \ldots\}$ and repeat the calculations seeking $\boldsymbol{\lambda}_{k+1\,l+1}$. To do this, the Newton–Raphson method (Problem 3.5.6), which is used to solve non-linear equations, will be applied.

The method is described as follows. For each $i \in I_{\mathrm{A}}(\boldsymbol{x}_{k+1})$, write $f_i\left(\boldsymbol{x}_k + \boldsymbol{y}_g\left(\boldsymbol{\lambda}_{k+1\,l}\right)\right)$ as $\bar{f}_i\left(\boldsymbol{\lambda}_{k+1\,l}\right)$. In the explanation of the Newton–Raphson method, a function $\boldsymbol{f}\left(\boldsymbol{x}_k + \boldsymbol{y}_g\right) = \boldsymbol{0}_{\mathbb{R}^d}$ of Eq. (3.5.7) with respect to $k \in \mathbb{N}$ was considered. Here, we shall consider $\left(\bar{f}_i\left(\boldsymbol{\lambda}_{k+1\,l} + \delta\boldsymbol{\lambda}\right)\right)_{i \in I_{\mathrm{A}}} = \boldsymbol{0}_{\mathbb{R}^{|I_{\mathrm{A}}|}}$ where $l \in \{0, 1, 2, \ldots\}$. Moreover, by taking into account the fact that $\boldsymbol{y}_g\left(\boldsymbol{\lambda}_{k+1\,l}\right)$ is a first-order function of $\boldsymbol{\lambda}_{k+1\,l}$ defined by Eq. (3.7.11), consider $\left(\boldsymbol{g}_i\left(\boldsymbol{\lambda}_{k+1\,l}\right) \cdot \boldsymbol{y}_{gj}\left(\boldsymbol{\lambda}_{k+1\,l}\right)\right)_{(i,j) \in I_{\mathrm{A}}^2}$ instead of $\boldsymbol{G}(\boldsymbol{x}_k)$ of Eq. (3.5.7). In this case, the following can be obtained analogous to Eq. (3.5.8):

$$\begin{aligned} \delta\boldsymbol{\lambda} &= (\delta\lambda_j)_{j \in I_{\mathrm{A}}} \\ &= -\left(\boldsymbol{g}_i\left(\boldsymbol{\lambda}_{k+1\,l}\right) \cdot \boldsymbol{y}_{gj}\left(\boldsymbol{\lambda}_{k+1\,l}\right)\right)^{-1}_{(i,j) \in I_{\mathrm{A}}^2} \left(f_i\left(\boldsymbol{\lambda}_{k+1\,l}\right)\right)_{i \in I_{\mathrm{A}}}. \end{aligned} \tag{3.7.21}$$

Using $\delta\boldsymbol{\lambda}$ of Eq. (3.7.21), the value of $\boldsymbol{\lambda}_{k+1}$ is updated using the recursion $\boldsymbol{\lambda}_{k+1\,l+1} = \boldsymbol{\lambda}_{k+1\,l} + \delta\boldsymbol{\lambda}$. Furthermore, from Eq. (3.7.11), $\boldsymbol{y}_g$ is modified, replacing it by $\boldsymbol{y}_g\left(\boldsymbol{\lambda}_{k+1\,l+1}\right)$. As a result, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ is changed to $\boldsymbol{x}_{k+1\,l+1} = \boldsymbol{x}_k + \boldsymbol{y}_g\left(\boldsymbol{\lambda}_{k+1\,l+1}\right)$. This update is used in step (11) of Algorithm 3.7.6 which will be shown later.

Now, let us discuss a method to correctly modify $\boldsymbol{\lambda}_{k+1}$ with respect to Exercise 3.7.4.

Exercise 3.7.5 (Volume minimizing problem) Recall from Exercise 3.7.4 that $\tilde{f}_1\left(\boldsymbol{a}_{(1)}\right) = 0.019687$. Using Eq. (3.7.21), correctly adjust $\lambda_{1(1)} = 0.0727397$ and obtain a trial point $\boldsymbol{a}_{(1)[l]}$ such that $\tilde{f}_1\left(\boldsymbol{a}_{(1)[l]}\right) \leq 10^{-4}$ holds. In this

calculation, the adjusted value of λ_1, for each step number k, should be written as $\lambda_{1(k+1)[l]}$, where l denotes the step number for the adjustment procedure. □

Answer Consider Eq. (3.7.21) as the example problem. Applying this to Exercise 3.7.4, we get

$$\delta\lambda_1 = -\frac{\tilde{f}_1\left(\boldsymbol{a}_{(1)[l]}\right)}{\boldsymbol{g}_{1(0)[l]} \cdot \boldsymbol{b}_{g1(0)[l]}}. \tag{3.7.22}$$

When $l = 0$, $\boldsymbol{a}_{(1)[0]} = \boldsymbol{a}_{(1)}$, $\boldsymbol{g}_{1(0)[0]} = \boldsymbol{g}_{1(0)}$ and $\boldsymbol{b}_{g1(0)[0]} = \boldsymbol{b}_{g1(0)}$, giving us $\delta\lambda_1 = 0.000863807$. At this point, λ_1 should be updated to

$$\lambda_{1(1)[1]} = \lambda_1 + \delta\lambda_1 = 0.0736035.$$

If this $\lambda_{1(1)[1]}$ is brought in and Eq. (3.7.11) is used to calculate $\boldsymbol{b}_{g(0)}$, we will get

$$\boldsymbol{b}_{g(0)[1]} = (0.0105202, -0.0884995)^\top.$$

Using this search vector to update the design variables will yield

$$\boldsymbol{a}_{(1)[1]} = \boldsymbol{a}_{(1)} + \boldsymbol{b}_{g(0)[1]} = (0.526649, 0.711501)^\top.$$

These calculations imply that $\tilde{f}_1\left(\boldsymbol{a}_{(1)[1]}\right) = 0.00066837 > 10^{-4}$. Clearly, the permitted constraint is not satisfied. Then, let $l = 1$ and repeat the steps above. From Eq. (3.7.22), one obtains

$$\delta\lambda_1 = -\frac{\tilde{f}_1\left(\boldsymbol{a}_{(1)[1]}\right)}{\boldsymbol{g}_{1(0)[1]} \cdot \boldsymbol{b}_{g1(0)[1]}} = 0.000029326.$$

Here, λ_1 is updated to

$$\lambda_{1(1)[2]} = \lambda_{1(1)[1]} + \delta\lambda_1 = 0.0736328.$$

If $\lambda_{1(1)[2]}$ and Eq. (3.7.11) is used to recalculate $\boldsymbol{b}_{g(0)}$, then we get

$$\boldsymbol{b}_{g(0)[2]} = (0.0105202, -0.0884995)^\top.$$

If this search vector is used to update the design variables, then we will have

$$\boldsymbol{a}_{(1)[2]} = \boldsymbol{a}_{(1)} + \boldsymbol{b}_{g(0)[2]} = (0.526693, 0.711505)^\top.$$

At this point, $\tilde{f}_1\left(\boldsymbol{a}_{(1)[2]}\right) = 0.0000243138 \leq 10^{-4}$. □

Meanwhile, the following method can be used to address the issue stated in (iii). The Lagrange function of the original problem (Problem 3.1.1) can be defined by

$$\mathscr{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f_0(\boldsymbol{x}) + \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x})} \lambda_i f_i(\boldsymbol{x}). \tag{3.7.23}$$

If with respect to $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$, $|f_i(\boldsymbol{x}_k)| \leq \epsilon_i$ is satisfied in order for $\mathscr{L}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) \approx f_0(\boldsymbol{x}_k)$ to hold, then the inequality

$$\epsilon_0 \gg \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{ki}\epsilon_i \tag{3.7.24}$$

must hold true. Hence, in order for this condition to hold, we require that a positive constant σ be sufficiently small relative to the unity, and that the criteria of constraint permissible values satisfy the following relation:

$$\epsilon_i \leq \frac{\sigma\epsilon_0}{|I_{\mathrm{A}}(\boldsymbol{x}_k)|\,\lambda_{ki}}, \tag{3.7.25}$$

for all $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$. If there is a case when there is an index i for which this condition does not hold, the criteria can be satisfied by substituting in a value smaller than $\sigma\epsilon_0/(|I_{\mathrm{A}}(\boldsymbol{x}_k)|\,\lambda_{ki})$ in ϵ_i. Such criteria for constraint concerning permissible values are used in Step (12) of Algorithm 3.7.6 that will be shown later.

On the other hand, a method to determine the step size $\|\boldsymbol{y}_g\|_X$ (in other words, c_a) so that the Armijo and Wolfe criteria are satisfied with respect to the Lagrange function can be thought of in connection with (iv) above. Theorem 3.4.7 became the basis to guarantee global convergence for unconstrained problems. Here it is assumed that the KKT conditions for the Lagrange multipliers and inequality constraints are satisfied at $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ from (ii) and (iii) above (in Algorithm 3.7.6, which will be shown later, the KKT conditions for the Lagrange multipliers and inequality constraints are satisfied after the step size is adjusted). Here, the Lagrange function matches f_0 and it becomes possible to use Armijo and Wolfe criteria with respect to unconstrained problems. Let us describe this more formally as follows. Let the Lagrange function of the original problem (Program 3.1.1) be Eq. (3.7.23). Let the gradient of $f_0(\boldsymbol{x}_k)$, $f_{i_1}(\boldsymbol{x}_k)$, $\ldots$, $f_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k)$ be written as $\boldsymbol{g}_0(\boldsymbol{x}_k)$, $\boldsymbol{g}_{i_1}(\boldsymbol{x}_k)$, $\ldots$, $\boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k)$, respectively, and the gradient of $f_0(\boldsymbol{x}_k + \boldsymbol{y}_g)$, $f_{i_1}(\boldsymbol{x}_k + \boldsymbol{y}_g)$, $\ldots$, $f_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k + \boldsymbol{y}_g)$ as $\boldsymbol{g}_0(\boldsymbol{x}_k + \boldsymbol{y}_g)$, $\boldsymbol{g}_{i_1}(\boldsymbol{x}_k + \boldsymbol{y}_g)$, $\ldots$, $\boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k + \boldsymbol{y}_g)$ as well. Here, the Armijo criterion with respect to $\xi \in (0,1)$ is given by

$$\begin{aligned}&\mathscr{L}(\boldsymbol{x}_k + \boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}) - \mathscr{L}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)\\ &\quad\leq \xi\left(\boldsymbol{g}_0(\boldsymbol{x}_k) + \sum_{i\in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{ki}\boldsymbol{g}_i(\boldsymbol{x}_k)\right)\cdot \boldsymbol{y}_g.\end{aligned} \tag{3.7.26}$$

Moreover, the Wolfe criterion with respect to μ $(0 < \xi < \mu < 1)$ is given by

$$\begin{aligned}&\mu\left(\boldsymbol{g}_0(\boldsymbol{x}_k) + \sum_{i\in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{ki}\boldsymbol{g}_i(\boldsymbol{x}_k)\right)\cdot \boldsymbol{y}_g\\ &\quad\leq \left(\boldsymbol{g}_0(\boldsymbol{x}_k + \boldsymbol{y}_g) + \sum_{i\in I_{\mathrm{A}}(\boldsymbol{x}_{k+1})} \lambda_{i\,k+1}\boldsymbol{g}_i(\boldsymbol{x}_k + \boldsymbol{y}_g)\right)\cdot \boldsymbol{y}_g.\end{aligned} \tag{3.7.27}$$

These criteria are used in steps (8) and (10) of Algorithm 3.7.6 which is shown below.

An example of an algorithm including the method such as the one above is shown next. Figure 3.17 shows its flow diagram.

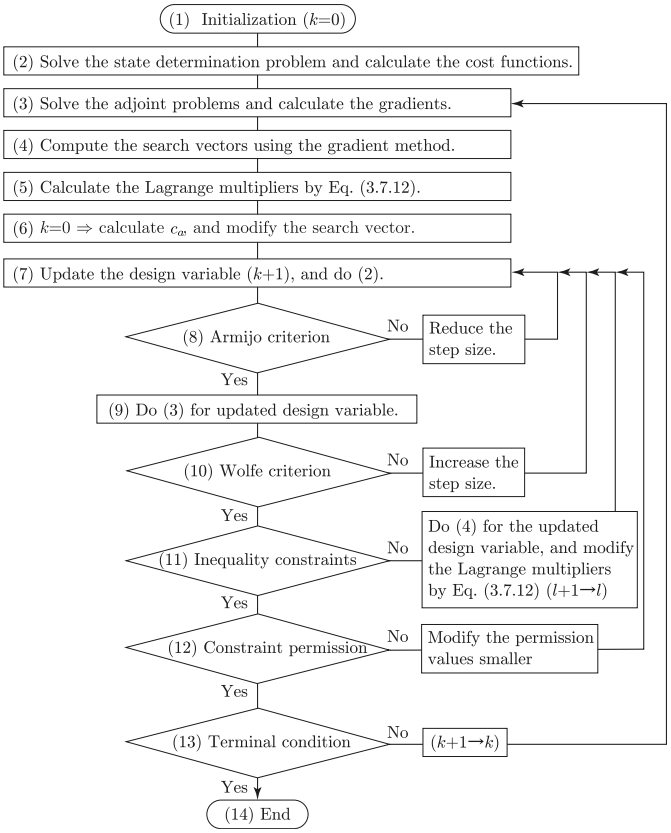


Fig. 3.17: Algorithm for the gradient method with respect to constraint problems with parameter adjustment.

Algorithm 3.7.6 (Gradient method with parameter adjustment)
Obtain the local minimum point of Problem 3.1.1 in the following way:

(1) Determine the initial point $\boldsymbol{x}_0$ so that $f_1(\boldsymbol{x}_0) \leq 0, \ldots, f_m(\boldsymbol{x}_0) \leq 0$ are satisfied. Also, determine the positive definite matrix $\boldsymbol{A}$ of Eq. (3.7.10) and initial step size ϵ_g (or the objective function reduce rate α), convergence check value ϵ_0 for f_0, initial values $\epsilon_1, \ldots, \epsilon_m$ of permissible ranges for $f_1, \ldots, f_m$, and Armijo and Wolfe standard values ξ and μ $(0 < \xi < \mu < 1)$ as well as the standard value σ $(\ll 1)$ of the constraint permissible values. Moreover, let $c_a = 1$, and set $k = l = 0$.

(2) Solve the state determination problem and calculate $f_0(\boldsymbol{x}_k)$, $f_1(\boldsymbol{x}_k)$, $\ldots$, $f_m(\boldsymbol{x}_k)$. Moreover, define

$$I_{\mathrm{A}}(\boldsymbol{x}_k) = \{\, i \in \{1, \ldots, m\} \mid f_i(\boldsymbol{x}_k) \geq -\epsilon_i \,\}.$$

(3) Solve the adjoint problem with respect to f_0, f_{i_1}, $\dots$, $f_{i_{|I_\mathrm{A}|}}$ and calculate $\boldsymbol{g}_0$, $\boldsymbol{g}_{i_1}$, $\dots$, $\boldsymbol{g}_{i_{|I_\mathrm{A}|}}$ at $\boldsymbol{x}_k$.

(4) Calculate $\boldsymbol{y}_{g0}$, $\boldsymbol{y}_{gi_1}$, $\dots$, $\boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$ using Eq. (3.7.10).

(5) Seek $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_{k+1\,l}$ using Eq. (3.7.12). If $I_\mathrm{I}(\boldsymbol{x}_k)$ in Eq. (3.7.9) is non-empty, replace $I_\mathrm{A}(\boldsymbol{x}_k) \setminus I_\mathrm{I}(\boldsymbol{x}_k)$ with $I_\mathrm{A}(\boldsymbol{x}_k)$ and solve Eq. (3.7.12) again.

(6) Obtain $\boldsymbol{y}_g$ with Eq. (3.7.11). When $k = 0$, let $\boldsymbol{y}_g = \bar{\boldsymbol{y}}_g$ and use Eq. (3.3.7) (or Eq. (3.3.9)) to obtain c_a. Moreover, for $i \in I_\mathrm{A}(\boldsymbol{x}_k)$, substitute $\bar{\boldsymbol{y}}_{gi}/c_a$ into $\boldsymbol{y}_{gi}$.

(7) Let $\boldsymbol{x}_{k+1\,l} = \boldsymbol{x}_k + \boldsymbol{y}_g(\boldsymbol{\lambda}_{k+1\,l})$ and calculate $f_0(\boldsymbol{x}_{k+1\,l})$, $f_1(\boldsymbol{x}_{k+1\,l})$, $\dots$, $f_m(\boldsymbol{x}_{k+1\,l})$. Moreover, define

$$I_\mathrm{A}(\boldsymbol{x}_{k+1}) = \{\, i \in \{1, \dots, m\} \mid f_i(\boldsymbol{x}_{k+1\,l}) \geq -\epsilon_i \,\}.$$

(8) Let $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_{k+1\,l}$ and check the Armijo criterion (Eq. (3.7.26)).

- If satisfied, proceed to the next step.
- Otherwise, let $\alpha > 1$, substitute αc_a into c_a and $\boldsymbol{y}_{g0}/c_a$, $\boldsymbol{y}_{gi_1}/c_a$, $\dots$, $\boldsymbol{y}_{gi_{|I_\mathrm{A}|}}/c_a$ into $\boldsymbol{y}_{g0}$, $\boldsymbol{y}_{gi_1}$, $\dots$, $\boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$ and then revert to (7).

(9) Calculate $\boldsymbol{g}_0$, $\boldsymbol{g}_{i_1}$, $\dots$, $\boldsymbol{g}_{i_{|I_\mathrm{A}|}}$ at $\boldsymbol{x}_{k+1}$.

(10) Let $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_{k+1\,l}$ and check the Wolfe criterion (Eq. (3.7.27)).

- If satisfied, proceed to the next step.
- Otherwise, let $\beta \in (0, 1)$ and substitute βc_a into c_a and $\beta\boldsymbol{y}_{g0}$, $\beta\boldsymbol{y}_{gi_1}$, $\dots$, $\beta\boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$ into $\boldsymbol{y}_{g0}$, $\boldsymbol{y}_{gi_1}$, $\dots$, $\boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$ and then return to (7).

(11) For $i \in I_\mathrm{A}(\boldsymbol{x}_{k+1})$, determine $|f_i(\boldsymbol{x}_{k+1})| \leq \epsilon_i$.

- If satisfied, proceed to the next step.
- Otherwise, at $\boldsymbol{x}_{k+1}$, calculate $\boldsymbol{g}_0$, $\boldsymbol{g}_{i_1}$, $\dots$, $\boldsymbol{g}_{i_{|I_\mathrm{A}|}}$ and also $\boldsymbol{y}_{g0}$, $\boldsymbol{y}_{gi_1}$, $\dots$, $\boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$ using Eq. (3.7.10), seek $\delta\boldsymbol{\lambda}$ with Eq. (3.7.21) and let $\boldsymbol{\lambda}_{k+1\,l+1} = \boldsymbol{\lambda}_{k+1\,l} + \delta\boldsymbol{\lambda}$. Afterwards, substitute $l+1$ into l and return to (7).

(12) For $i \in I_\mathrm{A}(\boldsymbol{x}_{k+1})$, check the criteria for the permissible values of the constraint (Eq. (3.7.25)).

- If satisfied, proceed to the next step.
- Otherwise, let $\beta \in (0, 1)$ with respect to unsatisfied i, substitute $\beta\sigma\epsilon_0 / (|I_\mathrm{A}(\boldsymbol{x}_{k+1})|\, \lambda_{i\,k+1})$ into ϵ_i and revert to (7).

(13) Check the terminal condition $|f_0(\boldsymbol{x}_{k+1}) - f_0(\boldsymbol{x}_k)| \leq \epsilon_0$.

- When the terminal condition is satisfied, proceed to the next step.
- Otherwise, substitute $k+1$ into k, let $l = 0$ and revert to (3).

(14) End the calculation.

□

3.8 Newton Method for Constrained Problems

If Hesse matrices of cost functions relating to the variation of $\boldsymbol{x}$ can be obtained, the Newton method can be used instead of the gradient method. We shall refer to this method as the Newton method for constrained problems. In this case, the Hesse matrices of $f_0, f_1, \ldots, f_m$ are expressed as $\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots, \boldsymbol{H}_m$, respectively. Moreover, if there is no confusion, $I_{\mathrm{A}}(\boldsymbol{x}_k)$ is written as I_{A}.

The Lagrange function $\mathscr{L}_{\mathrm{S}}$ of Eq. (3.7.3) defined with respect to the gradient method for constrained problems (Problem 3.7.1) is then replaced by

$$\begin{aligned}
&\mathscr{L}_{\mathrm{Q}}(\boldsymbol{y}, \boldsymbol{\lambda}_{k+1}) \\
&\quad = \frac{1}{2}\boldsymbol{y} \cdot (\boldsymbol{H}_0(\boldsymbol{x}_k)\,\boldsymbol{y}) + \boldsymbol{g}_0(\boldsymbol{x}_k) \cdot \boldsymbol{y} + f_0(\boldsymbol{x}_k) \\
&\qquad + \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x}_k)} \left\{ \lambda_{i\,k+1} (f_i + \boldsymbol{g}_i(\boldsymbol{x}_k) \cdot \boldsymbol{y}) + \lambda_{ik} \frac{1}{2} \boldsymbol{y} \cdot (\boldsymbol{H}_i(\boldsymbol{x}_k)\,\boldsymbol{y}) \right\}. \quad (3.8.1)
\end{aligned}$$

In the above formula, $\boldsymbol{\lambda}_k = (\lambda_{ik})_i$ denotes the Lagrange multiplier obtained in the previous step. When $k = 0$, it is supposed that it is determined via a method used in the gradient method for constrained problems. Let $\mathscr{L}_{\mathrm{S}}$ denote the Lagrange function in the next problem.

Problem 3.8.1 (Newton method for constrained problems) Let $\boldsymbol{x}_k \in X$ be a trial point in Problem 3.1.1, and $\boldsymbol{\lambda}_k \in \mathbb{R}^{|I_{\mathrm{A}}|}$ be the Lagrange multiplier satisfying Eq. (3.7.5) to Eq. (3.7.7). Moreover, let $f_0(\boldsymbol{x}_k)$, $f_{i_1}(\boldsymbol{x}_k) = 0, \ldots, f_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k) = 0$ as well as $\boldsymbol{g}_0(\boldsymbol{x}_k)$, $\boldsymbol{g}_{i_1}(\boldsymbol{x}_k), \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k)$ and $\boldsymbol{H}_0(\boldsymbol{x}_k)$, $\boldsymbol{H}_{i_1}(\boldsymbol{x}_k), \ldots, \boldsymbol{H}_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k)$ be known, and define

$$\boldsymbol{H}_{\mathscr{L}}(\boldsymbol{x}_k) = \boldsymbol{H}_0(\boldsymbol{x}_k) + \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{ik} \boldsymbol{H}_i(\boldsymbol{x}_k). \quad (3.8.2)$$

Under these assumptions, find $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ which satisfies

$$q(\boldsymbol{y}_g) = \min_{\boldsymbol{y} \in X} \left\{ q(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y} \cdot (\boldsymbol{H}_{\mathscr{L}}(\boldsymbol{x}_k)\,\boldsymbol{y}) + \boldsymbol{g}_0(\boldsymbol{x}_k) \cdot \boldsymbol{y} + f_0(\boldsymbol{x}_k) \;\middle|\; f_i(\boldsymbol{x}_k) + \boldsymbol{g}_i(\boldsymbol{x}_k) \cdot \boldsymbol{y} \leq 0 \text{ for } i \in I_{\mathrm{A}}(\boldsymbol{x}_k) \right\}.$$

□

Problem 3.8.1 can be classified as a second-order optimization problem. The expression $\boldsymbol{H}_{\mathscr{L}}(\boldsymbol{x}_k)$ need not be a positive definite real matrix, but if it is, Problem 3.8.1 is a convex optimization problem. Let us also consider the method for finding $\boldsymbol{y}_g$ using KKT conditions with respect to this problem.

The KKT conditions at minimum point $\boldsymbol{y}_g$ of Problem 3.8.1 are as follows:

$$\boldsymbol{H}_{\mathscr{L}}(\boldsymbol{x}_k)\boldsymbol{y}_g + \boldsymbol{g}_0(\boldsymbol{x}_k) + \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{i\,k+1}\boldsymbol{g}_i(\boldsymbol{x}_k) = \boldsymbol{0}_{X'}, \tag{3.8.3}$$

$$f_i(\boldsymbol{x}_{k+1}) = f_i(\boldsymbol{x}_k) + \boldsymbol{g}_i(\boldsymbol{x}_k)\cdot\boldsymbol{y}_g \leq 0 \quad \text{for } i \in I_{\mathrm{A}}(\boldsymbol{x}_k), \tag{3.8.4}$$

$$\lambda_{i\,k+1}\left(f_i(\boldsymbol{x}_k) + \boldsymbol{g}_i(\boldsymbol{x}_k)\cdot\boldsymbol{y}_g\right) = 0 \quad \text{for } i \in I_{\mathrm{A}}(\boldsymbol{x}_k), \tag{3.8.5}$$

$$\lambda_{i\,k+1} \geq 0 \quad \text{for } i \in I_{\mathrm{A}}(\boldsymbol{x}_k). \tag{3.8.6}$$

The pair $\left(\boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}\right) \in X \times \mathbb{R}^{|I_{\mathrm{A}}|}$ satisfying these conditions can be obtained in the following way. Suppose that those inequality constraints are active for all $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$. Then, Eq. (3.8.3) and Eq. (3.8.4) become

$$\begin{pmatrix} \boldsymbol{H}_{\mathscr{L}} & \boldsymbol{G}^{\top} \\ \boldsymbol{G} & \boldsymbol{0}_{\mathbb{R}^{|I_{\mathrm{A}}|\times|I_{\mathrm{A}}|}} \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_g \\ \boldsymbol{\lambda}_{k+1} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{g}_0 \\ (f_i)_{i\in I_{\mathrm{A}}} \end{pmatrix}, \tag{3.8.7}$$

where

$$\boldsymbol{G}^{\top} = \left(\boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}\right).$$

If $\boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$ are linearly independent and $\boldsymbol{H}_{\mathscr{L}}$ is regular, Eq. (3.8.7) becomes solvable around $\left(\boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}\right)$. With respect to these simultaneous first-order equations, define

$$I_{\mathrm{I}}(\boldsymbol{x}_k) = \{\, i \in I_{\mathrm{A}}(\boldsymbol{x}_k) \mid \lambda_{i\,k+1} < 0 \} \tag{3.8.8}$$

as the set of inactive constraint conditions, and when $I_{\mathrm{I}}(\boldsymbol{x}_k)$ is non-empty, replace $I_{\mathrm{A}}(\boldsymbol{x}_k) \setminus I_{\mathrm{I}}(\boldsymbol{x}_k)$ by $I_{\mathrm{A}}(\boldsymbol{x}_k)$ and then solve Eq. (3.8.7) again. The pair $\left(\boldsymbol{y}_g, \boldsymbol{\lambda}_{k+1}\right) \in X \times \mathbb{R}^{|I_{\mathrm{A}}|}$ obtained in this way satisfies Eq. (3.8.3) to Eq. (3.8.6).

Moreover, as also seen in the gradient method for constrained problems (Sect. 3.7), the following method can also be considered instead of directly solving the simultaneous first-order equations of Eq. (3.8.7). For each $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$ and with respect to $\boldsymbol{g}_i$, seek $\boldsymbol{y}_{g0}, \boldsymbol{y}_{gi_1}, \ldots, \boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}}$ so that the equation

$$\boldsymbol{y}_{gi} = -\boldsymbol{H}_{\mathscr{L}}^{-1}\boldsymbol{g}_i \tag{3.8.9}$$

holds. Furthermore, seek $\boldsymbol{\lambda}_{k+1}$ using Eq. (3.7.12). In this case, if $I_{\mathrm{I}}(\boldsymbol{x}_k)$ is non-empty, replace $I_{\mathrm{A}}(\boldsymbol{x}_k) \setminus I_{\mathrm{I}}(\boldsymbol{x}_k)$ by $I_{\mathrm{A}}(\boldsymbol{x}_k)$ and solve Eq. (3.7.8) again. From these results, $\boldsymbol{y}_g$ is obtained through Eq. (3.7.11).

The difference between this method and the gradient method for constrained problems is that $c_a \boldsymbol{A}$ is replaced by $\boldsymbol{H}_{\mathscr{L}}$. However, the search vector $\boldsymbol{y}_g$ obtained with this method can be expected to have the characteristics of the Newton method mentioned in Remark 3.5.4 because it uses the Hesse matrices of the cost functions. However, the following issues must be noted.

Remark 3.8.2 (Newton method for constrained problems) The cost function q of Problem 3.8.1 has the Hesse matrices of the constraint functions multiplied by the Lagrange multipliers of the previous step added on. As a result, the Hesse matrix is not used in the constraint conditions. From this, with respect to problems in which Lagrange multipliers satisfying the KKT conditions change a lot and for which the non-linearity of constraint functions is strong, there are cases when no convergence occurs unless the step size is made small enough. □

If the second-order derivative of a cost function is already obtained as the Hesse gradient, then the Newton method can now be illustrated as follows. In this case, Problem 3.8.1 is replaced with the following problem.

Problem 3.8.3 (Newton method using Hesse gradient) Let $X = \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ and c_a be a positive definite real symmetric matrix and a positive constant. Moreover, let $\boldsymbol{x}_k \in X$ be a trial point in Problem 3.1.1, and $\boldsymbol{\lambda}_k \in \mathbb{R}^{|I_{\mathrm{A}}|}$ be the Lagrange multiplier satisfying Eq. (3.7.5) to Eq. (3.7.7) (where $k+1$ replaces k). Furthermore, let $f_0(\boldsymbol{x}_k)$, $f_{i_1}(\boldsymbol{x}_k) = 0, \ldots, f_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k) = 0$ as well as $\boldsymbol{g}_0(\boldsymbol{x}_k)$, $\boldsymbol{g}_{i_1}(\boldsymbol{x}_k), \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k)$, a search vector $\bar{\boldsymbol{y}}_g$ and the Hesse gradients $\boldsymbol{g}_{\mathrm{H}0}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g)$, $\boldsymbol{g}_{\mathrm{H}i_1}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g), \ldots, \boldsymbol{g}_{\mathrm{H}i_{|I_{\mathrm{A}}|}}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g)$ be known, and define

$$\boldsymbol{g}_{\mathrm{H}\mathscr{L}}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g) = \boldsymbol{g}_{\mathrm{H}0}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g) + \sum_{i \in I_{\mathrm{A}}(\boldsymbol{x}_k)} \lambda_{ik} \boldsymbol{g}_{\mathrm{H}i}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g). \tag{3.8.10}$$

Under these assumptions, find $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ which satisfies

$$q(\boldsymbol{y}_g) = \min_{\boldsymbol{y} \in X} \Big\{ q(\boldsymbol{y}) = \frac{1}{2} \boldsymbol{y} \cdot (c_a \boldsymbol{A} \boldsymbol{y}) + \left(\boldsymbol{g}_0(\boldsymbol{x}_k) + \boldsymbol{g}_{\mathrm{H}\mathscr{L}}(\boldsymbol{x}_k, \bar{\boldsymbol{y}}_g) \right) \cdot \boldsymbol{y} + f_0(\boldsymbol{x}_k) \;\Big|\; f_i(\boldsymbol{x}_k) + \boldsymbol{g}_i(\boldsymbol{x}_k) \cdot \boldsymbol{y} \leq 0 \text{ for } i \in I_{\mathrm{A}}(\boldsymbol{x}_k) \Big\}$$

for all $\boldsymbol{y} \in X$. □

In solving Problem 3.8.3, particularly in the part where we employ the method using the search vectors obtained with respect to each cost function, the same algorithm with the Newton method can be applied by using

$$\boldsymbol{y}_{gi} = -(c_a \boldsymbol{A})^{-1} (\boldsymbol{g}_i + \boldsymbol{g}_{\mathrm{H}i}) \tag{3.8.11}$$

instead of Eq. (3.8.9).

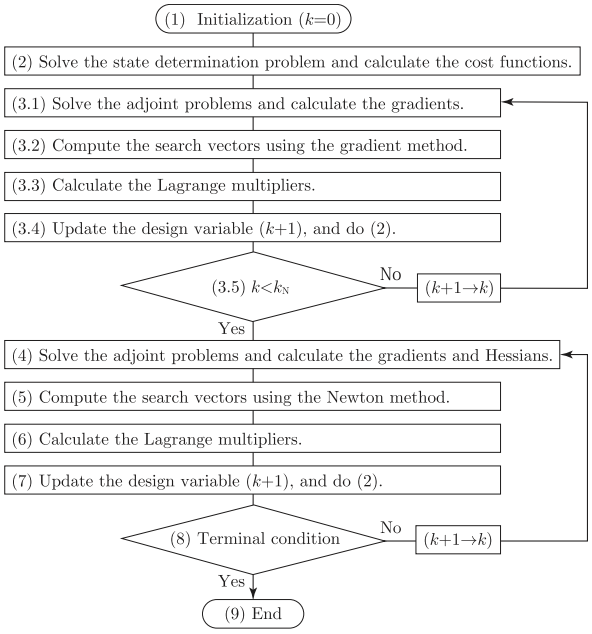


Fig. 3.18: Newton method algorithm with respect to constrained problems.

3.8.1 Simple Algorithm

Bearing in mind the ideas above, let us examine a simple algorithm based on the concept of the Newton method with respect to constrained problems. Figure 3.18 shows the flow diagram of the algorithm. Here, a method for seeking $\boldsymbol{y}_{gi}$ via Eq. (3.8.9) using $\boldsymbol{g}_i$ for every $i \in I_{\mathrm{A}}(\boldsymbol{x}_k)$ is used.

Algorithm 3.8.4 (Newton method for constrained problems) Obtain the local minimum point of Problem 3.1.1 in the following way:

(1) Determine the initial point $\boldsymbol{x}_0$ so that $f_1(\boldsymbol{x}_0) \leq 0, \ldots, f_m(\boldsymbol{x}_0) \leq 0$ are satisfied. Determine the positive constant ϵ_0 used for the convergence check of f_0 and the positive constants $\epsilon_1, \ldots, \epsilon_m$ which give the permissible range of $f_1, \ldots, f_m$. Set an iteration number k_{N} at which the Newton method starts, and $k = 0$.

(2) Solve the state determination problem and calculate $f_0(\boldsymbol{x}_k)$, $f_1(\boldsymbol{x}_k)$, $\ldots$, $f_m(\boldsymbol{x}_k)$. Moreover, define

$$I_{\mathrm{A}}(\boldsymbol{x}_k) = \{\, i \in \{1, \ldots, m\} \mid f_i(\boldsymbol{x}_k) \geq -\epsilon_i \,\}.$$

(3) Do the following when $k < k_{\mathrm{N}}$:

(3.1) Solve the adjoint problem with respect to $f_0, f_{i_1}, \ldots, f_{i_{|I_{\mathrm{A}}|}}$ and solve $\boldsymbol{g}_0, \boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$ at $\boldsymbol{x}_k$.

(3.2) Use Eq. (3.7.10) to calculate $\boldsymbol{y}_{g0}, \boldsymbol{y}_{gi_1}, \ldots, \boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$.

(3.3) Use Eq. (3.7.12) to seek $\boldsymbol{\lambda}_{k+1}$. If, in Eq. (3.7.9), $I_\mathrm{I}(\boldsymbol{x}_k)$ is non-empty, replace $I_\mathrm{A}(\boldsymbol{x}_k) \setminus I_\mathrm{I}(\boldsymbol{x}_k)$ with $I_\mathrm{A}(\boldsymbol{x}_k)$ and solve Eq. (3.7.12) again.

(3.4) Use Eq. (3.7.11) to seek $\boldsymbol{y}_g$ and let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ in order to calculate $f_0(\boldsymbol{x}_{k+1}), f_1(\boldsymbol{x}_{k+1}), \ldots, f_m(\boldsymbol{x}_{k+1})$. Moreover, let

$$I_\mathrm{A}(\boldsymbol{x}_{k+1}) = \{ i \in \{1, \cdots, m\} \mid f_i(\boldsymbol{x}_{k+1}) \geq -\epsilon_i \}.$$

(3.5) Substitute $k+1$ into k. When $k < k_\mathrm{N}$, revert to (3.1). Otherwise, proceed to (4).

(4) Solve the adjoint problem with respect to $f_0, f_{i_1}, \ldots, f_{i_{|I_\mathrm{A}|}}$ and calculate $\boldsymbol{g}_0, \boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_\mathrm{A}|}}$ and $\boldsymbol{H}_0, \boldsymbol{H}_{i_1}, \ldots, \boldsymbol{H}_{i_{|I_\mathrm{A}|}}$ at $\boldsymbol{x}_k$.

(5) Calculate $\boldsymbol{y}_{g0}, \boldsymbol{y}_{gi_1}, \ldots, \boldsymbol{y}_{gi_{|I_\mathrm{A}|}}$ using Eq. (3.8.9).

(6) Use Eq. (3.7.12) to seek $\boldsymbol{\lambda}_{k+1}$. If $I_\mathrm{I}(\boldsymbol{x}_k)$ in Eq. (3.7.9) is non-empty, replace $I_\mathrm{A}(\boldsymbol{x}_k) \setminus I_\mathrm{I}(\boldsymbol{x}_k)$ with $I_\mathrm{A}(\boldsymbol{x}_k)$ and solve Eq. (3.7.12) again.

(7) Use Eq. (3.7.11) to seek $\boldsymbol{y}_g$. Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{y}_g$ and solve the state determination problem in order to calculate $f_0(\boldsymbol{x}_{k+1}), f_1(\boldsymbol{x}_{k+1}), \ldots, f_m(\boldsymbol{x}_{k+1})$. Moreover, let

$$I_\mathrm{A}(\boldsymbol{x}_{k+1}) = \{ i \in \{1, \ldots, m\} \mid f_i(\boldsymbol{x}_{k+1}) \geq -\epsilon_i \}.$$

(8) Check the terminal condition $|f_0(\boldsymbol{x}_{k+1}) - f_0(\boldsymbol{x}_k)| \leq \epsilon_0$.

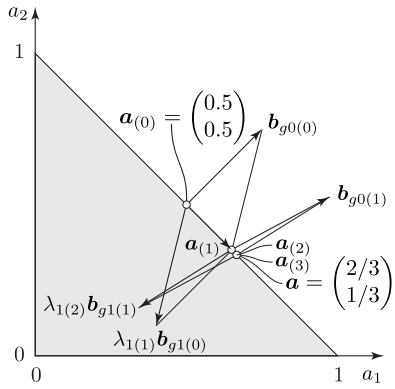
- When the terminal condition is satisfied, proceed to (9).
- Otherwise, substitute $k+1$ into k and revert to (4).

(9) End the calculations.

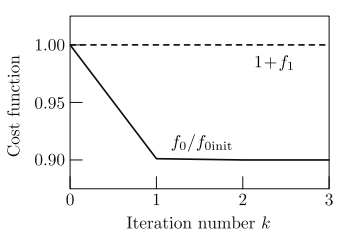
□

Let us use Algorithm 3.8.4 in order to find the trial points for Exercise 1.1.7 in Chap. 1.

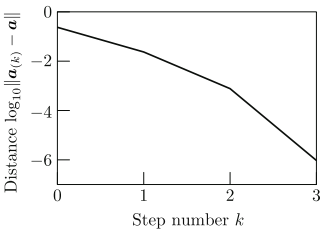
Exercise 3.8.5 (Mean compliance minimization problem) Consider Exercise 1.1.7. Let the initial point be $\boldsymbol{a}_{(0)} = (1/2, 1/2)^\top$ and use Algorithm 3.8.4 in order to obtain the trial points when $k \in \{0, 1\}$. Here, k_N should be taken as small as possible, and the numerical values required should be determined appropriately. □



(a) Movement of the trial point.



(b) History of cost function.



(c) Distance $\|\boldsymbol{a}_k - \boldsymbol{a}\|_{\mathbb{R}^2}$.

Fig. 3.19: Numerical example of mean compliance minimization problem using Newton method.

Answer The mean compliance $\tilde{f}_0(\boldsymbol{a})$ and the constraint function $f_1(\boldsymbol{a})$ with respect to volume are given respectively by Eq. (3.7.13) and Eq. (3.7.14). Moreover, their cross-sectional derivatives are given by Eq. (3.7.15) and Eq. (3.7.16). The Hesse matrix of $\tilde{f}_0(\boldsymbol{a})$ is

$$\boldsymbol{H}_0 = \begin{pmatrix} 8/a_1^3 & 0 \\ 0 & 2/a_2^3 \end{pmatrix}. \tag{3.8.12}$$

Moreover, $\boldsymbol{H}_1 = \boldsymbol{0}_{\mathbb{R}^{2\times 2}}$ and λ_1 is not required in Eq. (3.8.2), so we can take $k_{\mathrm{N}} = 0$. Let us seek numerical values using Algorithm 3.8.4. The design variable is again denoted by $\boldsymbol{a}_{(k)}$ for each step number k. The same is the case with $\boldsymbol{b}_{g0(k)}$, $\boldsymbol{b}_{g1(k)}$ and $\lambda_{1(k)}$.

(1) At the initial point $\boldsymbol{a}_{(0)} = (1/2, 1/2)^\top$, $f_1(\boldsymbol{a}_{(0)}) = 0$ is satisfied. Let $\epsilon_0 = 10^{-3}\tilde{f}_0(\boldsymbol{a}_{(0)})$, $\epsilon_1 = 10^{-6}$. Set $k = 0$.

(2) Equations (3.7.13) and (3.7.14) give $\tilde{f}_0(\boldsymbol{a}_{(0)}) = 10$ and $f_1(\boldsymbol{a}_{(0)}) = 0$, respectively. Let $I_{\mathrm{A}}(\boldsymbol{a}_{(0)}) = \{1\}$.

(3) Since $k = k_{\mathrm{N}}$, proceed to the next step.

(4) Equations (3.7.15) and (3.7.16) give $\boldsymbol{g}_{0(0)} = -(16, 4)^\top$, $\boldsymbol{g}_{1(0)} = (1, 1)^\top$.

(5) From Eq. (3.8.9), we get $\boldsymbol{b}_{g0(0)} = (1/4, 1/4)^\top$, $\boldsymbol{b}_{g1(0)} = -(1/64, 1/16)^\top$.

(6) Equation (3.7.12) gives $\boldsymbol{\lambda}_{1(1)} = 6.4$.

(7) Equation (3.7.11) gives $\boldsymbol{b}_{g(0)} = (0.15, -0.15)^\top$ and letting $\boldsymbol{a}_{(1)} = \boldsymbol{a}_{(0)} + \boldsymbol{b}_{g(0)} = (0.65, 0.35)^\top$ gives $\tilde{f}_0\left(\boldsymbol{a}_{(1)}\right) = 9.01099$, $f_1\left(\boldsymbol{a}_{(1)}\right) = 0$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(1)}\right) = \{1\}$.

(8) Since $\left|\tilde{f}_0\left(\boldsymbol{a}_{(1)}\right) - \tilde{f}_0\left(\boldsymbol{a}_{(0)}\right)\right| = 0.989011 \geq \epsilon_0 = 0.01$, the terminal condition is not satisfied. Then, set $k = 1$ and revert to (4).

(4) Equations (3.7.15) and (3.7.16) give $\boldsymbol{g}_{0(1)} = -(9.46746, 8.16327)^\top$, $\boldsymbol{g}_{1(1)} = (1, 1)^\top$.

(5) From Eq. (3.8.9), $\boldsymbol{b}_{g0(1)} = (0.325, 0.175)^\top$, $\boldsymbol{b}_{g1(1)} = -(0.0343281, 0.0214375)^\top$ can be obtained.

(6) Equation (3.7.12) is used to obtain $\boldsymbol{\lambda}_{1(2)} = 8.9661$.

(7) Equation (3.7.11) gives $\boldsymbol{b}_{g(1)} = (0.0172107, -0.0172107)^\top$, let $\boldsymbol{a}_{(2)} = \boldsymbol{a}_{(1)} + \boldsymbol{b}_{g(1)} = (0.667211, 0.332789)^\top$ which gives $\tilde{f}_0\left(\boldsymbol{a}_{(2)}\right) = 9.00001$, $f_1\left(\boldsymbol{a}_{(2)}\right) = 1.11022 \times 10^{-16}$. Moreover, let $I_{\mathrm{A}}\left(\boldsymbol{a}_{(1)}\right) = \{1\}$.

(8) $\left|\tilde{f}_0\left(\boldsymbol{a}_{(2)}\right) - \tilde{f}_0\left(\boldsymbol{a}_{(1)}\right)\right| = 0.010977 \geq \epsilon_0 = 0.01$ shows that the terminal condition is not satisfied, then substitute 2 into k and revert to (4).

Figure 3.19 shows these results and the succeding computed values. The calculation terminates at $k = 3$, since $\left|\tilde{f}_0\left(\boldsymbol{a}_{(3)}\right) - \tilde{f}_0\left(\boldsymbol{a}_{(2)}\right)\right| = 0.0000119968 \leq \epsilon_0 = 0.01$. □

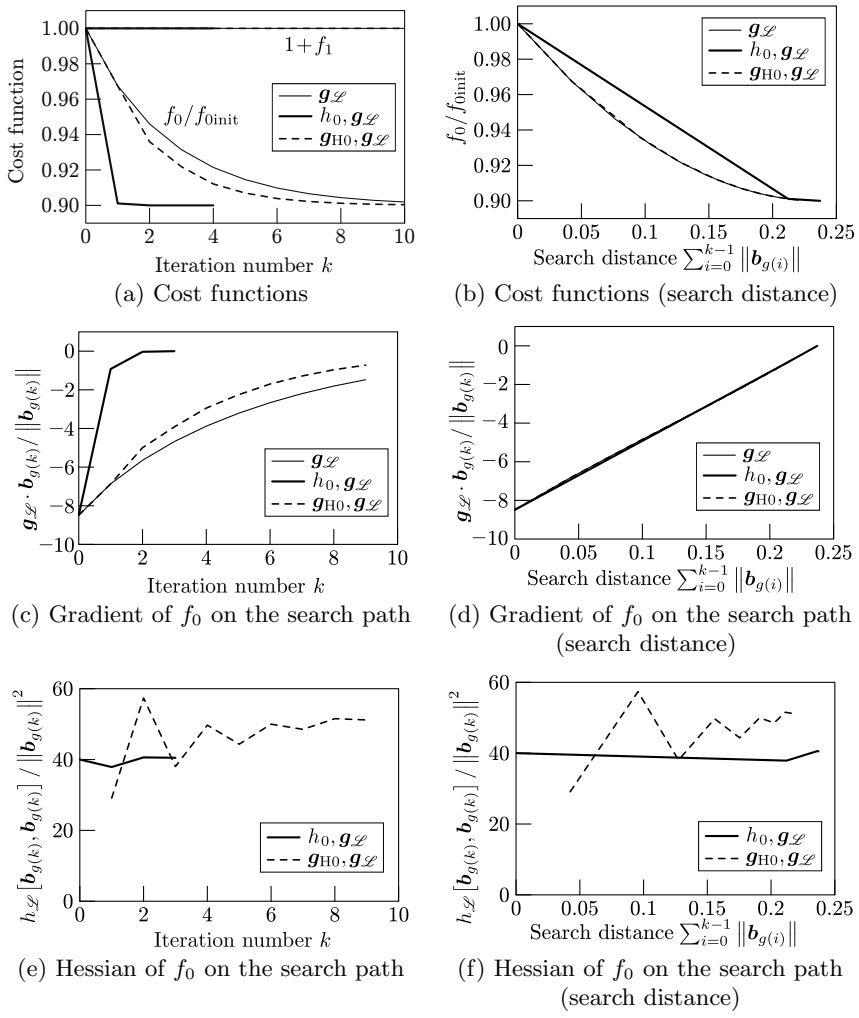
An algorithm via the Newton method using Hesse gradients for the second-order derivatives of cost functions is obtained by replacing Steps (4) and (5) in Algorithm 3.8.4 as follows:

(4) Solve the adjoint problems with respect to $f_0, f_{i_1}, \ldots, f_{i_{|I_{\mathrm{A}}|}}$ and calculate $\boldsymbol{g}_0, \boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{|I_{\mathrm{A}}|}}$. Moreover, solve the adjoint problems with respect to $f'_0, f'_{i_1}, \ldots, f'_{i_{|I_{\mathrm{A}}|}}$ and calculate $\boldsymbol{g}_{\mathrm{H}0}, \boldsymbol{g}_{\mathrm{H}i_1}, \ldots, \boldsymbol{g}_{\mathrm{H}i_{|I_{\mathrm{A}}|}}$.

(5) Calculate $\boldsymbol{y}_{g0}, \boldsymbol{y}_{gi_1}, \ldots, \boldsymbol{y}_{gi_{|I_{\mathrm{A}}|}}$ using Eq. (3.8.11).

Figures 3.20 and 3.21 show the result by the Newton method using the Hesse gradient $\boldsymbol{g}_{\mathrm{H}0}$ of f_0 in Exercise 1.1.7 from the initial point $\boldsymbol{a}_{(0)} = (1/2, 1/2)^\top$ together with the results by the gradient method (Exercise 3.7.3) and the Newton method (Exercise 3.8.5). In the gradient method, we set $\boldsymbol{A} = \boldsymbol{I}$ and the parameter value $c_a = 200$ was assumed. For the Newton method, using the Hesse gradient, we again set $\boldsymbol{A} = \boldsymbol{I}$ and chose $c_a = 200$ in the gradient method at $k = 0$ but took $c_a = 100$ for $k \geq k_{\mathrm{N}} = 1$.

Figure 3.20 (a) plots the cost functions $f_0/f_{0\mathrm{init}}$ and $1 + f_1$ normalized with f_0 at the initial shape denoted by $f_{0\mathrm{init}}$ and the volume at the initial shape denoted by $c_1 = 1$, respectively, at every iteration number k. Figure 3.20 (b) shows those values with respect to the distance $\sum_{i=0}^{k} \left\|\boldsymbol{b}_{g(i)}\right\|_X$ on the search path in $X = \mathbb{R}^2$. The graphs of f_0's gradient (the gradient of the Lagrange function $\mathscr{L} = \mathscr{L}_0 + \lambda_1 f_1$) calculated by $\boldsymbol{g}_{\mathscr{L}} \cdot \boldsymbol{b}_{g(k)} / \left\|\boldsymbol{b}_{g(k)}\right\|_X$ are shown in Fig. 3.20 (c) and (d) with respect to the iteration number and the search distance, respectively. Moreover, Fig. 3.20 (e) and (f) shows the graphs of f_0's second-order derivative



(a) Cost functions

(b) Cost functions (search distance)

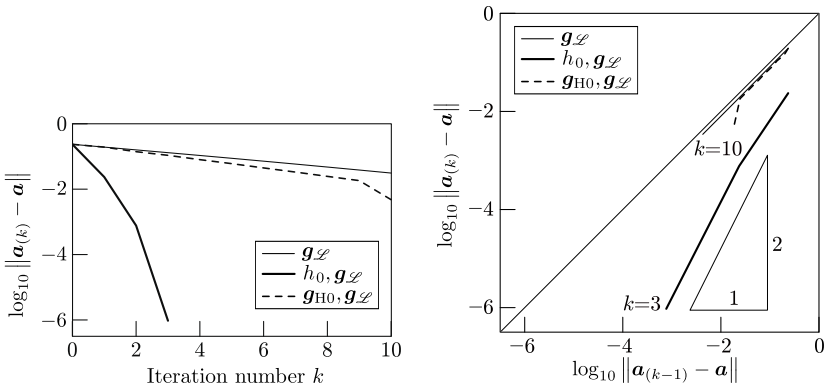
(c) Gradient of f_0 on the search path

(d) Gradient of f_0 on the search path (search distance)

(e) Hessian of f_0 on the search path

(f) Hessian of f_0 on the search path (search distance)

Fig. 3.20: Numerical example of mean compliance minimization: cost functions, gradients and Hessians of f_0 on the search path ($\boldsymbol{g}_{\mathscr{L}}$: Gradient method, $h_0, \boldsymbol{g}_{\mathscr{L}}$: Newton method, $\boldsymbol{g}_{\text{H}0}, \boldsymbol{g}_{\mathscr{L}}$: Newton method using Hesse gradient).



(a) Iteration history (b) $(k-1)$-th vs. k-th plot

Fig. 3.21: Numerical example of mean compliance minimization: error value $\|\boldsymbol{a}_k-\boldsymbol{a}\|_{\mathbb{R}^2}$ between the exact minimum point $\boldsymbol{a}$ and k-th approximation $\boldsymbol{a}_k$ ($\boldsymbol{g}_{\mathscr{L}}$: Gradient method, $h_0,\boldsymbol{g}_{\mathscr{L}}$: Newton method, $\boldsymbol{g}_{\mathrm{H0}},\boldsymbol{g}_{\mathscr{L}}$: Newton method using Hesse gradient).

$h_{\mathscr{L}}\left[\boldsymbol{b}_{g(k)},\boldsymbol{b}_{g(k)}\right]/\left\|\boldsymbol{b}_{g(k)}\right\|_X^2$ (in the case of the Newton method using the Hesse gradient, $\left(\boldsymbol{g}_{\mathrm{H0}}\cdot\boldsymbol{b}_{g(k)}+\lambda_1 h_1\left[\boldsymbol{b}_{g(k)},\boldsymbol{b}_{g(k)}\right]\right)/\left\|\boldsymbol{b}_{g(k)}\right\|_X^2=\boldsymbol{g}_{\mathrm{H0}}\cdot\boldsymbol{b}_{g(k)}/\left\|\boldsymbol{b}_{g(k)}\right\|_X^2$) with respect to the iteration number and the search distance, respectively.

From Fig. 3.20, it can be confirmed that the graphs with respect to the iteration number vary by the difference of the convergence speed, while the graphs with respect to the search distance are almost indistinguishable. The reason is that the search paths are the same as shown in Fig. 3.15 (a) and Fig. 3.19 (a). Such graphs will be shown in Chaps. 8 and 9, too. In these cases, however, it is quite difficult to obtain accurate plots of the search paths so they will no longer be illustrated graphically. Nevertheless, we want the reader to visualize them on their own.

In addition, Fig. 3.21 (a) shows the graphs of the error-norm $\|\boldsymbol{a}_k-\boldsymbol{a}\|_X$ between the minimum point $\boldsymbol{a}$ and the k-th approximation $\boldsymbol{a}_k$ obtained by the three methods. From this figure, it can be confirmed that the convergence order of the Newton method is higher than the first-order. Moreover, Fig. 3.21 (b) plots the k-th distance $\|\boldsymbol{a}_k-\boldsymbol{a}\|_X$ with respect to the $(k-1)$-th distance $\|\boldsymbol{a}_{k-1}-\boldsymbol{a}\|_X$. The indicated slopes of the graphs actually show the convergence order of each method, respectively. This is basically due to the fact that when the equation $\left\|\boldsymbol{a}_{(k)}-\boldsymbol{a}\right\|_X=r\left\|\boldsymbol{a}_{(k-1)}-\boldsymbol{a}\right\|_X^p$ is assumed, one also has the relation

$$\log_{10}\left\|\boldsymbol{a}_{(k)}-\boldsymbol{a}\right\|_X=p\log_{10}\left\|\boldsymbol{a}_{(k-1)}-\boldsymbol{a}\right\|_X+\log_{10}r. \tag{3.8.13}$$

From the above equation, it is clear that the gradient of the graph (or simply the slope of the graph) corresponds to the order p and the shift of the graph from the diagonal line corresponds to $\log_{10}r$. Based on the slopes of the above

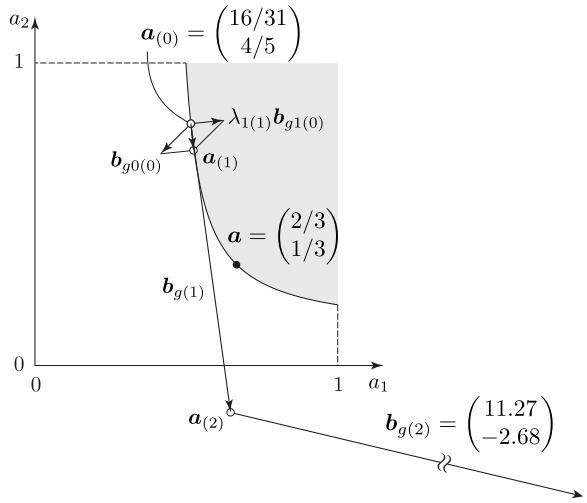


Fig. 3.22: Numerical example of a volume minimization problem using the Newton method.

plots, we can confirm that the convergence orders of the gradient and Newton method are of first and second order, respectively.

The above sample calculations confirm that the Newton method functions effectively with respect to mean compliance minimization problems. On the other hand, if the Newton method is used with respect to a volume minimization problems, such as Exercise 3.7.4, the resulting values will not reach convergence and will diverge instead. Figure 3.22 shows the movement of a trial point given by Algorithm 3.8.4. Here, for Lagrange multiplier $\lambda_{1(0)}$ when $k = 0$, $\lambda_{1(1)}$ in Exercise 3.7.4 was used. In this problem, the Hesse matrix of f_0 was $\mathbf{0}_{\mathbb{R}^{2\times 2}}$ and Hesse matrix of f_1 was positive definite. In other words, the conditions pointed out in Remark 3.8.2 are established. To make it possible to use the Newton method even in situations like this, there is a need to adjust the step size.

3.8.2 Complicated Algorithm

If the situation is such as that shown in Fig. 3.22 is considered, then there is a need to add a functionality for adjusting the step size as well as a functionality explained in Sect. 3.7.2. The relationship between these functionalities and algorithm are shown in Sect. 3.7.2, and so, we shall not repeat them here.

Meanwhile, if the Hesse matrix is not positive definite, there are known methods such as making it positive definite by adding a positive definite matrix or making it positive by removing the components of eigenmodes with negative eigenvalues. Furthermore, if it is not positive definite, a gradient method could be used in order to switch to the Newton method once it nears convergence.

3.9 Summary

Chapter 3 looked at methods for seeking the local minimum points with respect to non-linear optimization problems in finite-dimensional vector space. The key points are as follows:

(1) Iterative methods are used as standard techniques for solving non-linear optimization problems. An iterative method is one in which an initial point is provided and a trial point is updated while appropriately determining a search vector (search direction and step size) (Sect. 3.2).

(2) A representative method determining the search direction with respect to unconstrained optimization problem is the gradient method. This method is used for determining the search direction defined by the gradient of cost function with respect to the design variable (Sect. 3.3).

(3) With respect to unconstrained optimization problem, Armijo and Wolfe criteria are known as criteria for determinining the appropriateness of the step size. An iterative method, in which the step size has been decided in order to satisfy these criteria, has global convergence (Sect. 3.4).

(4) If the Hesse matrix and the gradient of the cost function with respect to an unconstrained optimization problem can be calculated, then by using a Newton method, the search direction and step size can be determined simultaneously. The trial point obtained via a Newton method converges quadratically. However, the calculation of the Hesse matrix can be costly (Sect. 3.5).

(5) The augmented function methods are known as a class of methods for solving optimization problems with inequality constraints. These methods are methods in which the constraint functions are multiplied by a constant representing weight and added to the objective function to make the problem an unconstrained one. However, in order to use these methods, there is a need to find an appropriate monotonic sequence of the constant for each problem (Sect. 3.6).

(6) A method for solving using KKT conditions can be considered in order to solve optimization problems with inequality constraints. If all the gradients of cost functions are computable, the gradient method with respect to constrained problems is used. In this method, the Lagrange multipliers are determined using the matrix constructed of search vectors which, on the other hand, are obtained through the gradient method using the gradients for each cost function, as well as the gradients themselves. This relationship is used effectively when considering a practical algorithm (Sect. 3.7).

(7) If the Hesse matrices of the cost functions in an optimization problem with inequality constraints can be calculated, the Newton method with

respect to constrained problems is used. In this method, the positive definite symmetric matrix used in the gradient method with respect to constrained problems is simply replaced by a Hesse matrix and the same algorithm is used as for the gradient method. In order for this method to function effectively, the non-linearity of the constrained functions should be weak (Sect. 3.8).

3.10 Practice Problems

3.1 Consider a problem seeking $x \in \mathbb{R}$ which satisfies the gradient $g(x) = 0$ with respect to the non-linear function $f \in C^2(\mathbb{R}; \mathbb{R})$. With respect to $k \in \mathbb{N}$, when $x_k \in \mathbb{R}$ and $g(x_k)$ are given, show the equation for obtaining x_{k+1} with the Newton–Raphson method (Problem 3.5.6). Moreover, show the equation for obtaining x_{k+1} when replacing $g(x_k)$ with the difference

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

This demonstrates a formula for the secant method.

3.2 Consider using the secant method as an algorithm to solve Problem 3.4.1 (the strict line search method). When $\bar{\epsilon}_{gl}$ is given with respect to $l \in \{0, 1, 2, \ldots\}$, show the equation for obtaining $\bar{\epsilon}_{g\,l+1}$.

3.3 Check that the search direction $\bar{\boldsymbol{y}}_{g\,k+1}$ calculated in the equations from Eq. (3.4.8) to Eq. (3.4.12) shown as an example of a conjugate gradient method (Problem 3.4.10) is conjugate to $\bar{\boldsymbol{y}}_{gk}$.

3.4 Consider a problem seeking the design variable $\boldsymbol{a}$ (the length of two edges) in Practice **1.6** for which the cost function $f(\boldsymbol{a})$ (volume of a tetrahedron) is minimized. Obtain the search vector $\boldsymbol{b}$ using the Newton method when the initial value of the design variable $\boldsymbol{a}_0 = (a_{01}, a_{02})^\top$ is given.