# 最佳化方法與應用
# MA5037-*

# Chapter 18. Sequential Quadratic Programming

# Introduction

對於非線性受限優化來說，其中一種最有效的方法是通過解決二次子問題來生成步進量。這種順序二次規劃（SQP）方法既可以在 line search 框架中使用，也可以在 trust region 框架中使用，適用於小型或大型問題。與線性限制拉格朗日法（第 17 章）不同，該方法在大部分限制為線性時有效，而當解決具有顯著非線性限制的問題時，SQP 方法顯示出其優勢。

本章考慮的所有方法都是 active set 法；對這一章節來說，可能更能具體描述本章的標題是“用於非線性規劃的 active set 法”。在第 14 章中，指定教科書上有提到“用於非線性規劃的內點法”，這是處理具不等式限制優化問題的一種競爭性方法。

# Introduction

對於非線性受限優化來說，其中一種最有效的方法是通過解決二次子問題來生成步進量。這種順序二次規劃（SQP）方法既可以在 line search 框架中使用，也可以在 trust region 框架中使用，適用於小型或大型問題。與線性限制拉格朗日法（第 17 章）不同，該方法在大部分限制為線性時有效，而當解決具有顯著非線性限制的問題時，SQP 方法顯示出其優勢。

本章考慮的所有方法都是 active set 法；對這一章節來說，可能更能具體描述本章的標題是“用於非線性規劃的 active set 法”。在第 14 章中，指定教科書上有提到“用於非線性規劃的內點法”，這是處理具不等式限制優化問題的一種競爭性方法。

# Introduction

在 active set SQP 方法中有兩種類型：

1. 在 IQP 方法中，每次迭代都解決一個通用的不等式限制二次規劃問題，其雙重目標是計算一個步進量並（自動地同時）生成對最佳 active set 的估計。
2. EQP 方法將上述步進量與 active set 的計算 decouple。它們首先計算最佳 active set 的估計，然後解決一個等式限制的二次規劃問題以找到步進量。

在本章中，我們研究了 IQP 和 EQP 方法。

我們對 SQP 方法的開發分為兩個階段。首先，我們考慮 local SQP 法，這些方法啟發了 SQP 方法並允許我們在簡單的情況下介紹步進量的計算技術。其次，我們考慮實際的 line search 和 trust region 方法，這些方法能夠從遠端起始點實現收斂。在整個章節中，我們考慮了解決大問題的算法需求。

# Introduction

在 active set SQP 方法中有兩種類型：

1. 在 IQP 方法中，每次迭代都解決一個通用的不等式限制二次規劃問題，其雙重目標是計算一個步進量並（自動地同時）生成對最佳 active set 的估計。
2. EQP 方法將上述步進量與 active set 的計算 decouple。它們首先計算最佳 active set 的估計，然後解決一個等式限制的二次規劃問題以找到步進量。

在本章中，我們研究了 IQP 和 EQP 方法。

我們對 SQP 方法的開發分為兩個階段。首先，我們考慮 local SQP 法，這些方法啟發了 SQP 方法並允許我們在簡單的情況下介紹步進量的計算技術。其次，我們考慮實際的 line search 和 trust region 方法，這些方法能夠從遠端起始點實現收斂。在整個章節中，我們考慮了解決大問題的算法需求。

# §18.1 Local SQP Method

我們首先考慮只具等式限制的優化問題

$$\min f(x) \quad \text{subject to} \quad c(x) = 0, \tag{1}$$

其中 $f : \mathbb{R}^n \to \mathbb{R}$ 和 $c : \mathbb{R}^n \to \mathbb{R}^m$ 是光滑函數。SQP 方法的想法是在當前的迭代點 $x_k$ 對 (1) 進行建模，得到一個二次規劃 (QP) 子問題，然後使用這個子問題的 minimizer 來定義一個新的迭代點 $x_{k+1}$。SQP 方法的挑戰在於設計能生成良好步進量的二次子問題。也許最簡單的 SQP 方法的推導將其視為對 (1) 的 KKT 最優條件應用牛頓法的一個應用。

# §18.1 Local SQP Method

根據 $(31)_{12}$，我們知道這個問題的拉格朗日函數為

$$\mathcal{L}(x,\lambda) = f(x) - \lambda^{\mathrm{T}} c(x),$$

而等式限制問題 (1) 的一階（KKT）條件 $(32)_{12}$ 可以被寫成一個包含 $n+m$ 個未知數 $x$ 和 $\lambda$ 的 $n+m$ 個方程式的系統：

$$F(x,\lambda) \equiv \nabla_{(x,\lambda)}\mathcal{L}(x,\lambda) = \begin{bmatrix} \nabla f(x) - A(x)^{\mathrm{T}}\lambda \\ c(x) \end{bmatrix} = 0, \qquad (2)$$

其中 $A(x)$ 來表示限制函數的 Jacobian 矩陣，即

$$A(x)^{\mathrm{T}} = \left[\nabla c_1(x) \vdots \nabla c_2(x) \vdots \cdots \vdots \nabla c_m(x)\right],$$

在此 $c_i(x)$ 是向量 $c(x)$ 的第 $i$ 個分量。對於等式限制問題 (1) 的任何解 $(x_*, \lambda_*)$，只要 $A(x_*)$ 具有滿秩（即 LICQ 在 $x_*$ 成立），(2) 式就會成立（或是更精確地說，$F(x_*,\lambda_*) = 0$ 成立）。

# §18.1 Local SQP Method

一個解非線性方程組 (2) 的自然想法是使用牛頓法。方程組 (2) 對於 $x$ 和 $\lambda$ 的 Jacobian 為：

$$\nabla_{(x,\lambda)} F(x,\lambda) = \begin{bmatrix} \nabla^2_{xx}\mathcal{L}(x,\lambda) & -A(x)^{\mathrm{T}} \\ A(x) & 0 \end{bmatrix}.$$

因此，從迭代點 $(x_k, \lambda_k)$ 出發的牛頓步進量 $(p_k, p_\lambda)^{\mathrm{T}}$ 是 Newton-KKT 系統

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f_k + A_k^{\mathrm{T}}\lambda_k \\ -c_k \end{bmatrix} \tag{3}$$

的解，其中 $\nabla^2_{xx}\mathcal{L}_k \equiv \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$，而由牛頓步進量生成的下一迭代點為

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix}. \tag{4}$$

# §18.1 Local SQP Method

當 (3) 中的 KKT 矩陣是 non-singular 時，這個牛頓迭代是有意義的。我們在第 16 章中看到，如果在 $(x, \lambda) = (x_k, \lambda_k)$ 處滿足以下假設，則該矩陣是 non-singular。

**Assumptions 18.1**.

(a) Constraint Jacobian $A(x)$ 有滿秩；

(b) 矩陣 $\nabla^2_{xx}\mathcal{L}(x, \lambda)$ 在限制的 tangent space 上是正定的，亦即

$$d^{\mathrm{T}}\nabla^2_{xx}\mathcal{L}(x, \lambda)d > 0 \quad \text{for all } d \neq 0 \text{ such that } A(x)d = 0.$$

第一個假設是在第 12 章討論的 LICQ，我們在本章中假設此條件成立。第二個條件在 $(x, \lambda)$ 接近最優解 $(x_*, \lambda_*)$ 且在解處滿足二階充分條件時成立（定理 12.6 - 見下頁）。在這些假設下，牛頓迭代 (3)、(4) 可被證明是二次收斂的，並且在起始點足夠接近 $x_*$ 的情況下，它是解決只具等式限制優化問題的優秀算法。

# Theorem 12.6 – Second-Order Conditions

## Theorem (Second-Order Sufficient Conditions)

*Suppose that for some feasible point* $x_* \in \mathbb{R}^n$ *there is a Lagrange multiplier vector* $\lambda_*$ *such that the KKT conditions*

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = 0\,, \tag{32a}_{12}$$

$$c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E}, \tag{32b}_{12}$$

$$c_i(x_*) \geqslant 0 \quad \text{for all } i \in \mathcal{I}, \tag{32c}_{12}$$

$$\lambda_i^* \geqslant 0 \quad \text{for all } i \in \mathcal{I}, \tag{32d}_{12}$$

$$\lambda_i^* c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}. \tag{32e}_{12}$$

*are satisfied. Suppose also that*

$$w^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}(x_*, \lambda_*) w > 0 \quad \text{for all } w \in \mathcal{C}(x_*, \lambda_*) \backslash \{0\}\,. \tag{60}_{12}$$

*Then* $x_*$ *is a strict local solution for*

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}\,, \\ c_i(x) \geqslant 0, i \in \mathcal{I}\,. \end{cases} \tag{1}_{12}$$

# §18.1 Local SQP Method

- **SQP framework**

還有另一種審視牛頓迭代 (3)、(4) 的方式。假設在迭代 $(x_k, \lambda_k)$ 時，我們使用二次規劃

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{5a}$$

$$\text{subject to}$$

$$A_k p + c_k = 0 \tag{5b}$$

求取此迭代的步進量。如果 Assumptions 18.1 成立，則該問題有一個滿足以下條件的唯一解 $(p_k, \ell_k)$：

$$\begin{aligned} \nabla_{xx}^2 \mathcal{L}_k p_k + \nabla f_k - A_k^{\mathrm{T}} \ell_k &= 0, \\ A_k p_k + c_k &= 0, \end{aligned}$$

其中 $\ell_k$ 為受限優化問題 (5) 的 Lagrange 乘子。

# §18.1 Local SQP Method

向量 $p_k$ 和 $\ell_k$ 可與牛頓方程組 (3) 的解相關：如果我們從 (3) 中的第一個方程的兩側減去 $A_k^{\mathrm{T}}\lambda_k$，我們得到

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{6}$$

因此，由於係數矩陣的可逆性，我們有 $\lambda_{k+1} = \ell_k = \lambda_k + p_\lambda$，且 $p_k$ 是 (3) 和 (5) 的解。

因此，新的迭代 $(x_{k+1}, \lambda_{k+1})$ 可以定義為二次規劃 (5) 的解，也可以定義為牛頓方法 (3)、(4) 應用於問題的最優性條件所生成的迭代。這兩種觀點都是有用的。牛頓觀點有助於分析，而 SQP 框架使我們能夠推導出實用的算法，並將技術擴展到不等式限制的情況下。

# §18.1 Local SQP Method

向量 $p_k$ 和 $\ell_k$ 可與牛頓方程組 (3) 的解相關：如果我們從 (3) 中的第一個方程的兩側減去 $A_k^{\mathrm{T}}\lambda_k$，我們得到

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{6}$$

因此，由於係數矩陣的可逆性，我們有 $\lambda_{k+1} = \ell_k = \lambda_k + p_\lambda$，且 $p_k$ 是 (3) 和 (5) 的解。

因此，新的迭代 $(x_{k+1}, \lambda_{k+1})$ 可以定義為二次規劃 (5) 的解，也可以定義為牛頓方法 (3)、(4) 應用於問題的最優性條件所生成的迭代。這兩種觀點都是有用的。牛頓觀點有助於分析，而 SQP 框架使我們能夠推導出實用的算法，並將技術擴展到不等式限制的情況下。

# §18.1 Local SQP Method

現在我們以最簡單的形式陳述 SQP 方法。

**Algorithm 18.1** (Local SQP Algorithm for solving (1)).

Choose an initial pair $(x_0, \lambda_0)$; set $k \leftarrow 0$;

**repeat** until a convergence test is satisfied

Evaluate $f_k$, $\nabla f_k$, $\nabla^2_{xx}\mathcal{L}_k$, $c_k$, and $A_k$;

Solve (5) to obtain $p_k$ and $\ell_k$;

Set $x_{k+1} \leftarrow x_k + p_k$ and $\lambda_{k+1} \leftarrow \ell_k$;

**end** (repeat)

# §18.1 Local SQP Method

注意到在二次規劃

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{5a}$$

$$\text{subject to}$$

$$A_k p + c_k = 0 \tag{5b}$$

中，我們可以將 (5a) 中的線性項 $\nabla f_k^{\mathrm{T}} p$ 替換為 $\nabla_x \mathcal{L}(x_k, \lambda_k)^{\mathrm{T}} p$，因為限制條件 (5b) 使得

$$\nabla_x \mathcal{L}(x_k, \lambda_k)^{\mathrm{T}} p = \nabla f_k^{\mathrm{T}} p - \lambda_k^{\mathrm{T}} A_k p = \nabla f_k^{\mathrm{T}} p + \lambda_k^{\mathrm{T}} c_k$$

因而這兩種線性化的選擇等價。在這種情況下，(5a) 是對拉格朗日函數的二次逼近。這個觀察為我們選擇二次模型 (5) 提供了動機：我們首先將非線性規劃 (1) 替換為對滿足等式限制 $c(x) = 0$ 的 Lagrangian 的最小化問題，然後對 Lagrangian 進行二次逼近，並對限制進行線性逼近以獲得 (5)。

# §18.1 Local SQP Method

注意到在二次規劃

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{5a}$$

$$\text{subject to}$$

$$A_k p + c_k = 0 \tag{5b}$$

中，我們可以將 (5a) 中的線性項 $\nabla f_k^{\mathrm{T}} p$ 替換為 $\nabla_x \mathcal{L}(x_k, \lambda_k)^{\mathrm{T}} p$，因為限制條件 (5b) 使得

$$\nabla_x \mathcal{L}(x_k, \lambda_k)^{\mathrm{T}} p = \nabla f_k^{\mathrm{T}} p - \lambda_k^{\mathrm{T}} A_k p = \nabla f_k^{\mathrm{T}} p + \lambda_k^{\mathrm{T}} c_k$$

因而這兩種線性化的選擇等價。在這種情況下，(5a) 是對拉格朗日函數的二次逼近。這個觀察為我們選擇二次模型 (5) 提供了動機：我們首先將非線性規劃 (1) 替換為對滿足等式限制 $c(x) = 0$ 的 Lagrangian 的最小化問題，然後對 Lagrangian 進行二次逼近，並對限制進行線性逼近以獲得 (5)。

# §18.1 Local SQP Method

- **Inequality constraints**

SQP 框架可以輕鬆地擴展到一般的非線性規劃問題

$$\min f(x) \tag{7a}$$

subject to

$$c_i(x) = 0, i \in \mathcal{E}, \tag{7b}$$

$$c_i(x) \geqslant 0, i \in \mathcal{I}. \tag{7c}$$

將問題 (7) 中的目標函數改以如 (5a) 中的二次函數逼近並將不等式和等式限制線性化，可得以下方程組

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{8a}$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{8b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}. \tag{8c}$$

# §18.1 Local SQP Method

我們可以使用第 16 章中描述的任何一個二次規劃算法來解決問題 (8)：若 $p_k$ 和 $\lambda_{k+1}$ 是 (8) 的解和相應的 Lagrange 乘子，則新的迭代點 $(x_{k+1}, p_{k+1}) = (x_k + p_k, \lambda_{k+1})$。因此，對於 (8)，一個 local SQP 方法可由 Algorithm 18.1 中將步進量改由 (8) 計算修改而得。

**Algorithm 18.1'** (Local SQP Algorithm for solving (7)).

Choose an initial pair $(x_0, \lambda_0)$; set $k \leftarrow 0$;

**repeat** until a convergence test is satisfied

Evaluate $f_k$, $\nabla f_k$, $\nabla^2_{xx}\mathcal{L}_k$, $c_k$, and $A_k$;

Solve (8) to obtain $p_k$ and $\ell_k$;

Set $x_{k+1} \leftarrow x_k + p_k$ and $\lambda_{k+1} \leftarrow \ell_k$;

**end** (repeat)

# §18.1 Local SQP Method

我們可以使用第 16 章中描述的任何一個二次規劃算法來解決問題 (8)：若 $p_k$ 和 $\lambda_{k+1}$ 是 (8) 的解和相應的 Lagrange 乘子，則新的迭代點 $(x_{k+1}, p_{k+1}) = (x_k + p_k, \lambda_{k+1})$。因此，對於 (8)，一個 local SQP 方法可由 Algorithm 18.1 中將步進量改由 (8) 計算修改而得。

**Algorithm 18.1'** (Local SQP Algorithm for solving (7)).

Choose an initial pair $(x_0, \lambda_0)$; set $k \leftarrow 0$;

**repeat** until a convergence test is satisfied

Evaluate $f_k$, $\nabla f_k$, $\nabla^2_{xx}\mathcal{L}_k$, $c_k$, and $A_k$;

Solve (8) to obtain $p_k$ and $\ell_k$;

Set $x_{k+1} \leftarrow x_k + p_k$ and $\lambda_{k+1} \leftarrow \ell_k$;

**end** (repeat)

# §18.1 Local SQP Method

在這個 IQP 方法中，問題

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{8a}$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{8b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}. \tag{8c}$$

的解之 active set $\mathcal{A}_k$（其與 $(x_k, \lambda_k)$ 有關因而加下標 $k$，並注意到 $\mathcal{A}_k \neq \mathcal{A}(x_k)$）構成我們對非線性規劃問題

$$\min f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geqslant 0, i \in \mathcal{I}. \end{cases} \tag{7}$$

的解之 active set 的猜測。如果 SQP 方法能夠正確識別出這個最佳的 active set（並且在後續迭代中不改變其猜測），那麼它將表現得像一個用於只具等式限制優化的牛頓方法，並且會迅速收斂。下頁的結果給出了這種理想行為發生的條件。

# §18.1 Local SQP Method

在這個 IQP 方法中，問題

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p \tag{8a}$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{8b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}. \tag{8c}$$

的解之 active set $\mathcal{A}_k$（其與 $(x_k, \lambda_k)$ 有關因而加下標 $k$，並注意到 $\mathcal{A}_k \neq \mathcal{A}(x_k)$）構成我們對非線性規劃問題

$$\min f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geqslant 0, i \in \mathcal{I}. \end{cases} \tag{7}$$

的解之 active set 的猜測。如果 SQP 方法能夠正確識別出這個最佳的 active set（並且在後續迭代中不改變其猜測），那麼它將表現得像一個用於只具等式限制優化的牛頓方法，並且會迅速收斂。下頁的結果給出了這種理想行為發生的條件。

# §18.1 Local SQP Method

## Theorem

*Let $x_*$ be a local solution of (7) at which the KKT conditions are satisfied for some $\lambda_*$. Suppose that at $(x_*, \lambda_*)$, it holds*

1. *the linear independence constraint qualification (LICQ); that is, $[\nabla c_i(x_*)]_{i\in\mathcal{A}(x_*)}$ has full rank;*
2. *the strict complementarity condition; that is,*
$$(\forall\, i \in \mathcal{I})((\lambda_i^* \neq 0) \vee (c_i(x_*) \neq 0));$$
3. *the second-order sufficient condition; that is,*
$$w^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}(x_*, \lambda_*) w > 0 \quad \text{for all } w \in \mathcal{C}(x_*, \lambda_*)\backslash\{0\}.$$

*If $(x_k, \lambda_k)$ is sufficiently close to $(x_*, \lambda_*)$, there is a local solution of the sub-problem (8) whose active set $\mathcal{A}_k$ is the same as the active set $\mathcal{A}(x_*)$ of the nonlinear program (7) at $x_*$.*

值得注意的是，在遠離解的地方 SQP 方法通常能夠改善 active set 的估計並引導迭代朝向解前進；請見 §18.7。

# §18.1 Local SQP Method

## Theorem

*Let $x_*$ be a local solution of (7) at which the KKT conditions are satisfied for some $\lambda_*$. Suppose that at $(x_*, \lambda_*)$, it holds*

1. *the linear independence constraint qualification (LICQ); that is, $[\nabla c_i(x_*)]_{i\in\mathcal{A}(x_*)}$ has full rank;*
2. *the strict complementarity condition; that is,*
$$(\forall\, i \in \mathcal{I})((\lambda_i^* \neq 0) \vee (c_i(x_*) \neq 0));$$
3. *the second-order sufficient condition; that is,*
$$w^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}(x_*, \lambda_*) w > 0 \quad \text{for all } w \in \mathcal{C}(x_*, \lambda_*) \backslash \{0\}.$$

*If $(x_k, \lambda_k)$ is sufficiently close to $(x_*, \lambda_*)$, there is a local solution of the sub-problem (8) whose active set $\mathcal{A}_k$ is the same as the active set $\mathcal{A}(x_*)$ of the nonlinear program (7) at $x_*$.*

值得注意的是，在遠離解的地方 SQP 方法通常能夠改善 active set 的估計並引導迭代朝向解前進；請見 §18.7。

# §18.2 Preview of Practical SQP Methods

- **IQP and EQP**

有兩種設計 SQP 方法來解決一般非線性規劃問題 (7) 的方式。第一種是上一節所描述的 IQP（不等式限制 QP）方法：該方法是在每次迭代中解決一個二次子問題 (8)，將此子問題的解（在該子問題中所對應的 active set）作為最優 active set 的猜測，並在實踐中已被證明是相當成功的方法。它的主要缺點是解決一般二次規劃問題 (8) 的計算量可能很大，特別是當問題規模很大時。然而，當 SQP 方法的迭代收斂到解時，如果我們善用前一次迭代的信息來對當前子問題的最優解做出良好的猜測，那麼解決二次子問題的計算成本將變得更低。下面描述了這種熱啟動策略。

# §18.2 Preview of Practical SQP Methods

第二種方法在每次迭代中選擇一個限制子集作為所謂的工作集 (working set)，並僅解決形式為

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{5a}$$

subject to

$$A_k p + c_k = 0 \tag{5b}$$

的只具等式限制的優化子問題，其中工作集中的限制被施加為等式，而所有其它限制被忽略。工作集通過基於拉格朗日乘子估計的規則更新，或通過解決輔助子問題進行每次迭代。這種 EQP（等式限制 QP）方法具有優勢，即在大規模情況下，只具等式限制的二次子問題比 (8) 更容易解決。

# §18.2 Preview of Practical SQP Methods

一個 EQP 方法的示例是在 §18.5 中討論的順序線性二次規劃（SLQP）方法。該方法通過從

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{8a}$$

中省略二次項 $p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p$ 並添加一個信任區域限制 $\|p\|_\infty \leqslant \Delta_k$ 到子問題來構造一個線性規劃問題。得到的線性規劃子問題的 active set 被視為當前迭代的工作集，然後固定工作集中的限制並解決一個只具等式限制的二次規劃問題（將項 $p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p$ 重新插入）以獲得 SQP 步進量。另一種成功的 EQP 方法是在 §16.7 中所描述的適用於「有界限限制 (bound constraint) 的二次規劃問題」的梯度投影方法。在這個方法中，工作集是通過「將最速下降方向投影在可行區域上獲得的路徑」上最小化二次模型來確定的。

# §18.2 Preview of Practical SQP Methods

一個 EQP 方法的示例是在 §18.5 中討論的順序線性二次規劃（SLQP）方法。該方法通過從

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{8a}$$

中省略二次項 $p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p$ 並添加一個信任區域限制 $\|p\|_\infty \leqslant \Delta_k$ 到子問題來構造一個線性規劃問題。得到的線性規劃子問題的 active set 被視為當前迭代的工作集，然後固定工作集中的限制並解決一個只具等式限制的二次規劃問題（將項 $p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p$ 重新插入）以獲得 SQP 步進量。另一種成功的 EQP 方法是在 §16.7 中所描述的適用於「有界限限制 (bound constraint) 的二次規劃問題」的梯度投影方法。在這個方法中，工作集是通過「將最速下降方向投影在可行區域上獲得的路徑」上最小化二次模型來確定的。

# §18.2 Preview of Practical SQP Methods

一個 EQP 方法的示例是在 §18.5 中討論的順序線性二次規劃（SLQP）方法。該方法通過從

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{8a}$$

中省略二次項 $p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p$ 並添加一個信任區域限制 $\|p\|_\infty \leqslant \Delta_k$ 到子問題來構造一個線性規劃問題。得到的線性規劃子問題的 active set 被視為當前迭代的工作集，然後固定工作集中的限制並解決一個只具等式限制的二次規劃問題（將項 $p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p$ 重新插入）以獲得 SQP 步進量。另一種成功的 EQP 方法是在 §16.7 中所描述的適用於「有界限限制 (bound constraint) 的二次規劃問題」的梯度投影方法。在這個方法中，工作集是通過「將最速下降方向投影在可行區域上獲得的路徑」上最小化二次模型來確定的。

# §18.2 Preview of Practical SQP Methods

- **Enforcing convergence**

為了實際應用，SQP 方法必須能夠從遠端起點和非凸問題上收斂。接下來我們概述如何調整 local SQP 策略以實現這些目標。

我們首先將其與無受限優化作比較。在無受限優化的最簡單形式中，優化函數 $f$ 的牛頓迭代給出二次模型

$$m_k(p) = f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 f_k p$$

的 minimizer。此框架在 $x_k$ 接近解是有用的，因為此時 Hessian 矩陣 $\nabla^2 f(x_k)$ 通常是正定的而使得上述二次模型有一個良好定義的極小值。然而當 $x_k$ 不接近解時，模型函數 $m_k$ 可能不是凸函數。Trust-region 方法通過將候選步進量 $p_k$ 限制在原點的某個鄰域內，確保新的迭代始終是良好定義且有用的。Line search 方法則是修改 $m_k(p)$ 中的 Hessian 矩陣，使其正定（可能用 quasi-Newton 逼近 $B_k$ 替換），以確保 $p_k$ 是目標函數 $f$ 的下降方向。

# §18.2 Preview of Practical SQP Methods

- **Enforcing convergence**

為了實際應用，SQP 方法必須能夠從遠端起點和非凸問題上收斂。接下來我們概述如何調整 local SQP 策略以實現這些目標。

我們首先將其與無受限優化作比較。在無受限優化的最簡單形式中，優化函數 $f$ 的牛頓迭代給出二次模型

$$m_k(p) = f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 f_k p$$

的 minimizer。此框架在 $x_k$ 接近解是有用的，因為此時 Hessian 矩陣 $\nabla^2 f(x_k)$ 通常是正定的而使得上述二次模型有一個良好定義的極小值。然而當 $x_k$ 不接近解時，模型函數 $m_k$ 可能不是凸函數。Trust-region 方法通過將候選步進量 $p_k$ 限制在原點的某個鄰域內，確保新的迭代始終是良好定義且有用的。Line search 方法則是修改 $m_k(p)$ 中的 Hessian 矩陣，使其正定（可能用 quasi-Newton 逼近 $B_k$ 替換），以確保 $p_k$ 是目標函數 $f$ 的下降方向。

# §18.2 Preview of Practical SQP Methods

類似的策略也被用於 global SQP 方法。如果在 active constraint 的切空間上，$\nabla^2_{xx}\mathcal{L}_k$ 是正定的，則二次子問題

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k p \tag{5a}$$

subject to

$$A_k p + c_k = 0 \tag{5b}$$

有唯一解。當 $\nabla^2_{xx}\mathcal{L}_k$ 不具有此性質時，line search 方法要麼用正定逼近 $B_k$ 替換它，要麼在矩陣分解過程中直接修改 $\nabla^2_{xx}\mathcal{L}_k$。在所有這些情況下，子問題 (5) 有解，但修改可能會在模型中引入不需要的扭曲。

# §18.2 Preview of Practical SQP Methods

Trust-region SQP 方法在子問題中增加了一個將步進量限制在一個區域內的限制，其中無論 $\nabla^2_{xx}\mathcal{L}_k$ 是否正定模型 (5) 都被認為是可靠的模型。這些方法能夠處理 indefinite Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$。然而，trust region 的引入可能會使子問題變得不可行，即使子問題可行處理這種情況的程序會使演算法變得複雜並增加計算量。由於這些權衡，目前對於這兩種 SQP 方法中的任何一種 – line search 或 trust-region – 都沒有被明確地認為優於另一種。

用於接受或拒絕步進量的技術也影響了 SQP 方法的效率。在無受限優化中，merit function 簡單地是目標函數 $f$，並且在整個最小化過程中保持不變。對於受限優化問題，我們使用 merit function 或 filter 等裝置（見 §15.4）。這些裝置中使用的參數或條目必須以與 SQP 方法產生的步進量相容的方式進行更新。

# §18.2 Preview of Practical SQP Methods

Trust-region SQP 方法在子問題中增加了一個將步進量限制在一個區域內的限制，其中無論 $\nabla^2_{xx}\mathcal{L}_k$ 是否正定模型 (5) 都被認為是可靠的模型。這些方法能夠處理 indefinite Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$。然而，trust region 的引入可能會使子問題變得不可行，即使子問題可行處理這種情況的程序會使演算法變得複雜並增加計算量。由於這些權衡，目前對於這兩種 SQP 方法中的任何一種 – line search 或 trust-region – 都沒有被明確地認為優於另一種。

用於接受或拒絕步進量的技術也影響了 SQP 方法的效率。在無受限優化中，merit function 簡單地是目標函數 $f$，並且在整個最小化過程中保持不變。對於受限優化問題，我們使用 merit function 或 filter 等裝置（見 §15.4）。這些裝置中使用的參數或條目必須以與 SQP 方法產生的步進量相容的方式進行更新。

# §18.2 Preview of Practical SQP Methods

Trust-region SQP 方法在子問題中增加了一個將步進量限制在一個區域內的限制，其中無論 $\nabla^2_{xx}\mathcal{L}_k$ 是否正定模型 (5) 都被認為是可靠的模型。這些方法能夠處理 indefinite Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$。然而，trust region 的引入可能會使子問題變得不可行，即使子問題可行處理這種情況的程序會使演算法變得複雜並增加計算量。由於這些權衡，目前對於這兩種 SQP 方法中的任何一種 – line search 或 trust-region – 都沒有被明確地認為優於另一種。

用於接受或拒絕步進量的技術也影響了 SQP 方法的效率。在無受限優化中，merit function 簡單地是目標函數 $f$，並且在整個最小化過程中保持不變。對於受限優化問題，我們使用 merit function 或 filter 等裝置（見 §15.4）。這些裝置中使用的參數或條目必須以與 SQP 方法產生的步進量相容的方式進行更新。

# §18.2 Preview of Practical SQP Methods

Trust-region SQP 方法在子問題中增加了一個將步進量限制在一個區域內的限制，其中無論 $\nabla^2_{xx}\mathcal{L}_k$ 是否正定模型 (5) 都被認為是可靠的模型。這些方法能夠處理 indefinite Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$。然而，trust region 的引入可能會使子問題變得不可行，即使子問題可行處理這種情況的程序會使演算法變得複雜並增加計算量。由於這些權衡，目前對於這兩種 SQP 方法中的任何一種 – line search 或 trust-region – 都沒有被明確地認為優於另一種。

用於接受或拒絕步進量的技術也影響了 SQP 方法的效率。在無受限優化中，merit function 簡單地是目標函數 $f$，並且在整個最小化過程中保持不變。對於受限優化問題，我們使用 merit function 或 filter 等裝置（見 §15.4）。這些裝置中使用的參數或條目必須以與 SQP 方法產生的步進量相容的方式進行更新。

# §18.3 Algorithmic Development

在本節中，我們擴展了上一節的思想，描述了製造實用的 SQP 演算法所需的各種要素。我們重點關注確保子問題始終可行的技術，對二次模型的 Hessian 矩陣的替代選擇，以及步進量接受機制。

- **Handling inconsistent linearizations**

SQP 方法可能面臨的一個困難是非線性限制的線性化

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{8b}$$

和

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}. \tag{8c}$$

可能導致一個不可行的子問題。例如，考慮當 $n = 1$ 且限制為 $x \leqslant 1$ 和 $x^2 \geqslant 4$ 時。當我們在 $x_k = 1$ 處對這些限制進行線性化時，我們得到不一致的不等式

$$-p \geqslant 0 \quad \text{and} \quad 2p - 3 \geqslant 0.$$

# §18.3 Algorithmic Development

在本節中，我們擴展了上一節的思想，描述了製造實用的 SQP 演算法所需的各種要素。我們重點關注確保子問題始終可行的技術，對二次模型的 Hessian 矩陣的替代選擇，以及步進量接受機制。

- **Handling inconsistent linearizations**

SQP 方法可能面臨的一個困難是非線性限制的線性化

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{8b}$$

和

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}. \tag{8c}$$

可能導致一個不可行的子問題。例如，考慮當 $n = 1$ 且限制為 $x \leqslant 1$ 和 $x^2 \geqslant 4$ 時。當我們在 $x_k = 1$ 處對這些限制進行線性化時，我們得到不一致的不等式

$$-p \geqslant 0 \quad \text{and} \quad 2p - 3 \geqslant 0.$$

# §18.3 Algorithmic Development

為了克服這個困難，我們可以將非線性問題

$$\min f(x) \tag{7a}$$

subject to

$$c_i(x) = 0, i \in \mathcal{E}, \tag{7b}$$

$$c_i(x) \geqslant 0, i \in \mathcal{I}. \tag{7c}$$

重新定義為 $\ell_1$ 懲罰問題：

$$\min_{x,v,w,t} f(x) + \mu \sum_{i \in \mathcal{E}} (v_i + w_i) + \mu \sum_{i \in \mathcal{I}} t_i \tag{9a}$$

subject to

$$c_i(x) = v_i - w_i, i \in \mathcal{E}, \tag{9b}$$

$$c_i(x) \geqslant -t_i, i \in \mathcal{I}, \tag{9c}$$

$$v, w, t \geqslant 0, \tag{9d}$$

其中 $\mu > 0$ 是懲罰參數。

# §18.3 Algorithmic Development

此 $\ell_1$ 懲罰問題 (9) 總是可行的。如第 17 章所討論的，如果非線性問題 (7) 具有滿足某些正則性假設的解 $x_*$，並且如果懲罰參數 $\mu$ 足夠大，那麼 $x_*$（以及對於等式限制的 index $i$ 有 $v_i^* = w_i^* = 0$ 以及對不等式限制的 index $i$ 有 $t_i^* = 0$）是懲罰問題 (9) 的解。另一方面，如果非線性問題沒有可行解且 $\mu$ 足夠大，那麼懲罰問題通常確定一個不可行度量的 stationary point。$\mu$ 的選擇在第 17 章已經討論過，並且將在 §18.5 中再次討論。SNOPT 套件使用了公式 (9)，有時被稱為彈性模式 (elastic mode)，來處理線性化限制的不一致性。

在與 trust-region 方法相關的內容上，§18.5 還介紹了其他放寬限制的程序。

# §18.3 Algorithmic Development

此 $\ell_1$ 懲罰問題 (9) 總是可行的。如第 17 章所討論的，如果非線性問題 (7) 具有滿足某些正則性假設的解 $x_*$，並且如果懲罰參數 $\mu$ 足夠大，那麼 $x_*$（以及對於等式限制的 index $i$ 有 $v_i^* = w_i^* = 0$ 以及對不等式限制的 index $i$ 有 $t_i^* = 0$）是懲罰問題 (9) 的解。另一方面，如果非線性問題沒有可行解且 $\mu$ 足夠大，那麼懲罰問題通常確定一個不可行度量的 stationary point。$\mu$ 的選擇在第 17 章已經討論過，並且將在 §18.5 中再次討論。SNOPT 套件使用了公式 (9)，有時被稱為彈性模式 (elastic mode)，來處理線性化限制的不一致性。

在與 trust-region 方法相關的內容上，§18.5 還介紹了其他放寬限制的程序。

# §18.3 Algorithmic Development

- **Full quasi-Newton approximations**

Lagrangian 的 Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ 由目標函數和限制函數的二次導數組成。在某些應用中，這些信息不容易計算，因此考慮用 quasi-Newton 近似替換 (8a) 中的 Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ 是有用的。由於在無受限優化的背景下，BFGS 和 SR1 公式已經被證明是成功的，所以我們可以在這裡也使用它們。

從迭代的第 $k$ 步到第 $k+1$ 步，為使用 BFGS 或 SR1 公式（見第六章）來計算新的近似 $B_{k+1}$，我們必須利用向量

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1}). \quad (10)$$

我們可以將這個過程看作是「目標函數由 Lagrangian $\mathcal{L}(x, \lambda)$ 在 $\lambda$ 固定時給出」然後應用 quasi-Newton 更新的過程。這種觀點立即揭示了這種方法的優勢和劣勢。

# §18.3 Algorithmic Development

- **Full quasi-Newton approximations**

Lagrangian 的 Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ 由目標函數和限制函數的二次導數組成。在某些應用中，這些信息不容易計算，因此考慮用 quasi-Newton 近似替換 (8a) 中的 Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ 是有用的。由於在無受限優化的背景下，BFGS 和 SR1 公式已經被證明是成功的，所以我們可以在這裡也使用它們。

從迭代的第 $k$ 步到第 $k+1$ 步，為使用 BFGS 或 SR1 公式（見第六章）來計算新的近似 $B_{k+1}$，我們必須利用向量

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1}). \quad (10)$$

我們可以將這個過程看作是「目標函數由 Lagrangian $\mathcal{L}(x, \lambda)$ 在 $\lambda$ 固定時給出」然後應用 quasi-Newton 更新的過程。這種觀點立即揭示了這種方法的優勢和劣勢。

# §18.3 Algorithmic Development

如果在最小值發生的區域中 $\nabla^2_{xx}\mathcal{L}$ 是正定的，那麼使用 BFGS 更新的 $B_k$ 將反映問題的某些曲率信息，並且迭代將穩健且快速地收斂，如同無受限優化中的 BFGS 方法中一樣。然而，如果 $\nabla^2_{xx}\mathcal{L}$ 包含負特徵值，則使用正定矩陣近似它的 BFGS 方法可能會有問題。BFGS 更新要求 $s_k$ 和 $y_k$ 滿足曲率條件 $s_k^{\mathrm{T}} y_k > 0$，在 (10) 定義的情況下，即使迭代接近解，也可能無法滿足此條件。

為了克服這個困難，我們可以在不滿足條件

$$s_k^{\mathrm{T}} y_k \geqslant \theta s_k^{\mathrm{T}} B_k s_k$$

時跳過 BFGS 更新，在此 $\theta$ 是一個正參數（比如 $10^{-2}$）。這種策略有時可能會導致性能不佳甚至失敗，因此不能被視為通用算法的充分解決方案。

# §18.3 Algorithmic Development

如果在最小值發生的區域中 $\nabla^2_{xx}\mathcal{L}$ 是正定的，那麼使用 BFGS 更新的 $B_k$ 將反映問題的某些曲率信息，並且迭代將穩健且快速地收斂，如同無受限優化中的 BFGS 方法中一樣。然而，如果 $\nabla^2_{xx}\mathcal{L}$ 包含負特徵值，則使用正定矩陣近似它的 BFGS 方法可能會有問題。BFGS 更新要求 $s_k$ 和 $y_k$ 滿足曲率條件 $s_k^{\mathrm{T}} y_k > 0$，在 (10) 定義的情況下，即使迭代接近解，也可能無法滿足此條件。

為了克服這個困難，我們可以在不滿足條件

$$s_k^{\mathrm{T}} y_k \geqslant \theta s_k^{\mathrm{T}} B_k s_k$$

時跳過 BFGS 更新，在此 $\theta$ 是一個正參數（比如 $10^{-2}$）。這種策略有時可能會導致性能不佳甚至失敗，因此不能被視為通用算法的充分解決方案。

# §18.3 Algorithmic Development

一種更有效的可確保更新始終是良好定義的修改方法是通過修改 $y_k$ 的定義來實現的。

**Procedure 18.2** (Damped BFGS Updating).

Given: symmetric and positive definite matrix $B_k$;

Define $s_k$ and $y_k$ as in (10) and set

$$r_k = \theta_k y_k + (1 - \theta_k) B_k s_k,$$

where the scalar $\theta_k$ is defined as

$$\theta_k = \begin{cases} 1 & \text{if } s_k^{\mathrm{T}} y_k \geqslant 0.2 s_k^{\mathrm{T}} B_k s_k, \\ \dfrac{0.8 s_k^{\mathrm{T}} B_k s_k}{s_k^{\mathrm{T}} B_k s_k - s_k^{\mathrm{T}} y_k} & \text{if } s_k^{\mathrm{T}} y_k < 0.2 s_k^{\mathrm{T}} B_k s_k; \end{cases} \tag{11}$$

Update $B_k$ as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + \frac{r_k r_k^{\mathrm{T}}}{s_k^{\mathrm{T}} r_k}. \tag{12}$$

# §18.3 Algorithmic Development

公式 (12) 就是標準的 BFGS 更新公式，只是用 $r_k$ 替換了 $y_k$。這樣做保證了 $B_{k+1}$ 是正定的，因為可以證明當 $\theta_k \neq 1$ 時，我們有

$$s_k^{\mathrm{T}} r_k = 0.2 s_k^{\mathrm{T}} B_k s_k > 0.$$

為了對這種策略有更深入的理解，注意到若選擇 $\theta_k = 0$ 會得到 $B_{k+1} = B_k$，而選擇 $\theta_k = 1$ 則會得到未修改的 BFGS 更新產生的（可能 indefinite）矩陣。因此，在 $(0, 1)$ 中的 $\theta_k$ 的值會產生一個介於當前近似 $B_k$ 和未修改 BFGS 公式產生的近似之間的矩陣。選擇 $\theta_k$ 確保了新的近似保持足夠接近當前近似 $B_k$ 以確保其正定性。

# §18.3 Algorithmic Development

Damped BFGS 更新通常效果良好，但在困難問題上也可能表現不佳。它仍然無法解決 Lagrangian Hessian 可能不是正定的根本問題。出於這個原因，SR1 更新可能更為適合，確實是信任區域 SQP 方法的不錯選擇。通過應用公式 (6.24) 並使用 (10) 定義的 $s_k$ 和 $y_k$，使用第 6 章描述的保護措施，可以獲得對 Lagrangian 的 Hessian 的 SR1 近似。然而，line search 方法不能接受不定的 Hessian 近似，因此可能需要修改 SR1 公式，可能是通過添加單位矩陣的足夠大的倍數；請參見課本 (19.25) 附近的討論。

上述所有討論的 quasi-Newton 近似 $B_k$ 都是密集的 $n \times n$ 矩陣，在問題規模大的情況下可能會導致存儲和操作方面的高昂成本。在這種情況下，有限記憶體更新非常有用，並且通常在套件中實現（參見 (19.29) 中有限記憶體 BFGS 在受限優化算法中的實現）。

# §18.3 Algorithmic Development

Damped BFGS 更新通常效果良好，但在困難問題上也可能表現不佳。它仍然無法解決 Lagrangian Hessian 可能不是正定的根本問題。出於這個原因，SR1 更新可能更為適合，確實是信任區域 SQP 方法的不錯選擇。通過應用公式 (6.24) 並使用 (10) 定義的 $s_k$ 和 $y_k$，使用第 6 章描述的保護措施，可以獲得對 Lagrangian 的 Hessian 的 SR1 近似。然而，line search 方法不能接受不定的 Hessian 近似，因此可能需要修改 SR1 公式，可能是通過添加單位矩陣的足夠大的倍數；請參見課本 (19.25) 附近的討論。

上述所有討論的 quasi-Newton 近似 $B_k$ 都是密集的 $n \times n$ 矩陣，在問題規模大的情況下可能會導致存儲和操作方面的高昂成本。在這種情況下，有限記憶體更新非常有用，並且通常在套件中實現（參見 (19.29) 中有限記憶體 BFGS 在受限優化算法中的實現）。

# §18.3 Algorithmic Development

- **Reduced-Hessian quasi-Newton approximations**

當我們檢查對只具等式的受限優化問題 (1) 的 KKT 系統 (6) 時，我們會看到步進量 $p_k$ 在 $A_k^{\mathrm{T}}$ 的值域中的部分是完全由第二個區塊列 $A_k p_k = -c_k$ 決定的。Lagrangian Hessian $\nabla^2_{xx}\mathcal{L}_k$ 僅影響 $p_k$ 在正交子空間（即 $A_k$ 的 null space）中的部分。因此，考慮僅對影響 $p_k$ 在 $A_k$ 的 null space 中的分量的 $\nabla^2_{xx}\mathcal{L}_k$ 的那部分進行近似的 quasi-Newton 方法是合理的。在本節中，我們考慮基於這些 reduced-Hessian 近似的 quasi-Newton 方法。我們的重點在於只具等式的受限優化問題，因為現有的 SQP 方法對於完整問題 (7) 僅在生成等式限制子問題後使用 reduced-Hessian 方法。

# §18.3 Algorithmic Development

為了推導 reduced-Hessian 方法，我們考慮使用 §16.2 中的 null space 方法來解步進量方程式

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{6}$$

在該節中，我們定義了矩陣 $Y_k$ 和 $Z_k$，其列分別 span 了 $A_k^{\mathrm{T}}$ 的值域和 $A_k$ 的 null space。將步進量 $p_k$ 寫為 $p_k = Y_k p_Y + Z_k p_Z$ 並將其代入 (6) 式，我們獲得以下 $p_Y$ 和 $p_Z$ 滿足的系統：

$$(A_k Y_k) p_Y = -c_k, \tag{14a}$$

$$(Z_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k Z_k) p_Z = -Z_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k Y_k p_Y - Z_k^{\mathrm{T}} \nabla f_k. \tag{14b}$$

從方程組 (6) 的第一個 block 中我們可以看出（這個有時稱為 QP 乘子的 Lagrange 乘子）$\lambda_{k+1}$ 可以通過解以下方程式獲得：

$$(A_k Y_k)^{\mathrm{T}} \lambda_{k+1} = Y_k^{\mathrm{T}} (\nabla f_k + \nabla^2_{xx}\mathcal{L}_k p_k). \tag{13}$$

# §18.3 Algorithmic Development

為了推導 reduced-Hessian 方法，我們考慮使用 §16.2 中的 null space 方法來解步進量方程式

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{6}$$

在該節中，我們定義了矩陣 $Y_k$ 和 $Z_k$，其列分別 span 了 $A_k^{\mathrm{T}}$ 的值域和 $A_k$ 的 null space。將步進量 $p_k$ 寫為 $p_k = Y_k p_Y + Z_k p_Z$ 並將其代入 (6) 式，我們獲得以下 $p_Y$ 和 $p_Z$ 滿足的系統：

$$(A_k Y_k) p_Y = -c_k, \tag{14a}$$

$$(Z_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k Z_k) p_Z = -Z_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k Y_k p_Y - Z_k^{\mathrm{T}} \nabla f_k. \tag{14b}$$

從方程組 (6) 的第一個 block 中我們可以看出（這個有時稱為 QP 乘子的 Lagrange 乘子）$\lambda_{k+1}$ 可以通過解以下方程式獲得：

$$(A_k Y_k)^{\mathrm{T}} \lambda_{k+1} = Y_k^{\mathrm{T}} (\nabla f_k + \nabla^2_{xx}\mathcal{L}_k p_k). \tag{13}$$

# §18.3 Algorithmic Development

我們可以通過在 null space 方法中引入幾個近似來避免計算 Hessian $\nabla^2_{xx}\mathcal{L}_k$。首先注意到因為當迭代接近解時 $p_k$ 會收斂於零而 $\nabla f_k$ 通常不會收斂到零，我們可以從 (13) 式右側刪除包含 $p_k$ 的項，從而 decouple $p_k$ 和 $\lambda_{k+1}$ 的計算，並且消除了解 (13) 中對計算 $\nabla^2_{xx}\mathcal{L}_k$ 的需求。用這種方式計算的乘子將是解附近 QP 乘子的良好估計。更具體地說，如果我們選擇 $Y_k = A_k^{\mathrm{T}}$（當 $A_k$ 有滿秩時這是一個可行的選擇），我們可以得到：

$$\widehat{\lambda}_{k+1} = (A_k A_k^{\mathrm{T}})^{-1} A_k \nabla f_k. \tag{15}$$

這些 $\widehat{\lambda}_{k+1}$ 被稱為最小平方乘子，因為它們可以通過解

$$\min_{\lambda} \|\nabla_x \mathcal{L}(x_k, \lambda)\|^2 = \left\|\nabla f_k - A_k^{\mathrm{T}}\lambda\right\|^2$$

而獲得。

# §18.3 Algorithmic Development

這一觀察顯示，即使當前的迭代離解決方案很遠，最小平方乘子仍然是有用的，因為它們尋求盡可能地滿足 (2) 中的一階最優條件。從概念上講，使用最小平方乘子將 SQP 方法從 $x$ 和 $\lambda$ 的 primal-dual 迭代轉化為僅在 $x$ 變量中進行的純原始迭代。

# §18.3 Algorithmic Development

我們對 null space 方法的第二種簡化方法是刪除

$$(Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k)p_Z = \cancel{-Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k p_Y} - Z_k^{\mathrm{T}}\nabla f_k. \tag{14b}$$

中的交叉項 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k p_Y$，從而得到更簡單的系統：

$$(Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k)p_Z = -Z_k^{\mathrm{T}}\nabla f_k. \tag{16}$$

這種方法的優勢在於它只需要 $(n-m)\times(n-m)$ 近似矩陣 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k$，而不是 $(n-m)\times m$ 的交叉項矩陣 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k$，當 $m \gg n-m$ 時，這是一個相對較大的矩陣。當 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k$ 被一個 quasi-Newton 近似取代時，可以合理地放棄交叉項，因為法向分量 $p_Y$ 通常比切向分量 $p_Z$ 更快地收斂於零，從而使得 (16) 成為 (14b) 的一個良好的近似。

# §18.3 Algorithmic Development

我們對 null space 方法的第二種簡化方法是刪除

$$(Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k)p_Z = \cancel{-Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k p_Y} - Z_k^{\mathrm{T}}\nabla f_k. \tag{14b}$$

中的交叉項 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k p_Y$，從而得到更簡單的系統：

$$(Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k)p_Z = -Z_k^{\mathrm{T}}\nabla f_k. \tag{16}$$

這種方法的優勢在於它只需要 $(n-m)\times(n-m)$ 近似矩陣 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k$，而不是 $(n-m)\times m$ 的交叉項矩陣 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k$，當 $m \gg n-m$ 時，這是一個相對較大的矩陣。當 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k$ 被一個 quasi-Newton 近似取代時，可以合理地放棄交叉項，因為法向分量 $p_Y$ 通常比切向分量 $p_Z$ 更快地收斂於零，從而使得 (16) 成為 (14b) 的一個良好的近似。

# §18.3 Algorithmic Development

在放棄了部分 Hessian $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Y_k$ 之後，我們討論如何逼近剩餘部分 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k$。假設我們剛剛走了一個步進量 $\alpha_k p_k = x_{k+1} - x_k = \alpha_k Z_k p_Z + \alpha_k Y_k p_Y$。定義 $\nabla_{xx}^2\mathcal{L}_{k+1} = \nabla_{xx}^2\mathcal{L}(x_{k+1}, \lambda_{k+1})$，則根據泰勒定理我們有

$$\nabla_{xx}^2\mathcal{L}_{k+1}\alpha_k p_k \approx \nabla_x\mathcal{L}(x_k + \alpha_k p_k, \lambda_{k+1}) - \nabla_x\mathcal{L}(x_k, \lambda_{k+1}).$$

等號左右兩邊的前面同乘上 $Z_k^{\mathrm{T}}$，我們得到

$$\begin{aligned} & Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_{k+1}Z_k\alpha_k p_Z \\ & \quad \approx -Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_{k+1}Y_k\alpha_k p_Y + Z_k^{\mathrm{T}}[\nabla_x\mathcal{L}(x_k + \alpha_k p_k, \lambda_{k+1}) \\ & \qquad -\nabla_x\mathcal{L}(x_k, \lambda_{k+1})]. \end{aligned} \tag{17}$$

# §18.3 Algorithmic Development

如果我們放棄交叉項 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_{k+1}Y_k\alpha_k p_Y$（使用之前討論的理由），並以下式定義 $s_k$ 與 $y_k$：

$$\begin{aligned} s_k &= \alpha_k p_Z, \\ y_k &= Z_k^{\mathrm{T}}[\nabla_x\mathcal{L}(x_k+\alpha_k p_k, \lambda_{k+1}) - \nabla_x\mathcal{L}(x_k, \lambda_{k+1})]. \end{aligned} \tag{18}$$

則 $M_k \equiv Z_{k-1}^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_{k-1}$ 滿足割線方程 (secand equation)

$$M_{k+1}s_k = y_k.$$

然後，我們可以使用這些定義的校正向量 $s_k$ 和 $y_k$ 在 BFGS 或 SR1 公式中，以定義新的近似 $M_{k+1}$。相較於完整 Hessian quasi-Newton 逼近，這種 reduced-Hessian 方法的優點之一是，即使當前迭代距離解答還有一段距離，reduced-Hessian 更有可能是正定的。當使用 BFGS 公式時，在 line search 的實現中將需要更少地使用上述討論的保護機制。

# §18.3 Algorithmic Development

- **Merit functions**

SQP 方法通常使用一個 merit function 來決定是否接受試探步驟。在 line search 方法中，merit function 控制步驟的大小；而在 trust-region 方法中，它用以確定步進量是被接受還是拒絕，以及信賴區域半徑是否應該調整。在 SQP 方法中，已經使用了包括非平滑懲罰函數和增廣 Lagrangian 等多種 merit function。接下來我們將討論限於 exact 非平滑的 merit function，這類 merit function 的代表是在第 15 章和第 17 章討論的 $\ell_1$ merit function。

# §18.3 Algorithmic Development

在接下來的討論中，我們假設所有限制都以等式的形式存在，因為不等式限制 $c(x) \geqslant 0$ 通常被轉換為以下形式：

$$\bar{c}(x, s) = c(x) - s = 0,$$

其中 $s \geqslant 0$ 是鬆弛 (slack) 向量（而不等式限制條件 $s \geqslant 0$ 通常不受 merit function 監控）。

對只具等式限制的優化問題 (1)，$\ell_1$ merit function 的形式如下：

$$\phi_1(x; \mu) = f(x) + \mu \|c(x)\|_1 . \tag{19}$$

在 line search 方法中，滿足充分減少條件

$$\phi_1(x_k + \alpha_k p_k; \mu_k) \leqslant \phi_1(x_k; \mu_k) + \eta \alpha_k D(\phi_1(x_k; \mu); p_k), \eta \in (0,1), \tag{20}$$

的步進量 $\alpha_k p_k$ 將被接受，這裡的 $D(\phi_1(x_k; \mu); p_k)$ 表示 $\phi_1$ 在方向 $p_k$ 上的方向導數。在下頁的定理中我們將證明：如果選擇的懲罰參數 $\mu$ 足夠大，則這個下降條件成立。

# §18.3 Algorithmic Development

在接下來的討論中，我們假設所有限制都以等式的形式存在，因為不等式限制 $c(x) \geqslant 0$ 通常被轉換為以下形式：

$$\bar{c}(x, s) = c(x) - s = 0,$$

其中 $s \geqslant 0$ 是鬆弛 (slack) 向量（而不等式限制條件 $s \geqslant 0$ 通常不受 merit function 監控）。

對只具等式限制的優化問題 (1)，$\ell_1$ merit function 的形式如下：

$$\phi_1(x; \mu) = f(x) + \mu \|c(x)\|_1 . \tag{19}$$

在 line search 方法中，滿足充分減少條件

$$\phi_1(x_k + \alpha_k p_k; \mu_k) \leqslant \phi_1(x_k; \mu_k) + \eta \alpha_k D(\phi_1(x_k; \mu); p_k), \eta \in (0, 1), \tag{20}$$

的步進量 $\alpha_k p_k$ 將被接受，這裡的 $D(\phi_1(x_k; \mu); p_k)$ 表示 $\phi_1$ 在方向 $p_k$ 上的方向導數。在下頁的定理中我們將證明：如果選擇的懲罰參數 $\mu$ 足夠大，則這個下降條件成立。

# §18.3 Algorithmic Development

## Theorem

*Let $p_k$ and $\lambda_{k+1}$ be generated by the SQP iteration*

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{6}$$

*Then the directional derivative of $\phi_1$ in the direction $p_k$ satisfies*

$$D(\phi_1(x_k;\mu);p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1. \tag{21}$$

*Moreover, we have that*

$$D(\phi_1(x_k;\mu);p_k) \leqslant -p_k^{\mathrm{T}}\nabla^2_{xx}\mathcal{L}_k p_k - (\mu - \|\lambda_{k+1}\|_\infty)\|c_k\|_1. \tag{22}$$

## Proof.

By applying Taylor's theorem to $f$ and $c_i$, $i = 1, 2, \cdots, m$, we obtain

$$\begin{aligned} &\phi_1(x_k + \alpha p;\mu) - \phi_1(x_k;\mu) \\ &\quad = f(x_k + \alpha p) - f_k + \mu\|c(x_k + \alpha p)\|_1 - \mu\|c_k\|_1 \\ &\quad \leqslant \alpha\nabla f_k^{\mathrm{T}} p + \gamma\alpha^2\|p\|^2 + \mu\|c_k + \alpha A_k p\|_1 - \mu\|c_k\|_1, \end{aligned}$$

where $\gamma$ bounds the second-derivative terms in $f$ and $c$. □

# §18.3 Algorithmic Development

## Theorem

*Let $p_k$ and $\lambda_{k+1}$ be generated by the SQP iteration*

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{6}$$

*Then the directional derivative of $\phi_1$ in the direction $p_k$ satisfies*

$$D(\phi_1(x_k;\mu);p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1. \tag{21}$$

*Moreover, we have that*

$$D(\phi_1(x_k;\mu);p_k) \leqslant -p_k^{\mathrm{T}}\nabla^2_{xx}\mathcal{L}_k p_k - (\mu - \|\lambda_{k+1}\|_\infty)\|c_k\|_1. \tag{22}$$

## Proof.

By applying Taylor's theorem to $f$ and $c_i$, $i = 1, 2, \cdots, m$, we obtain

$$\begin{aligned} &\phi_1(x_k + \alpha p;\mu) - \phi_1(x_k;\mu) \\ &\quad = f(x_k + \alpha p) - f_k + \mu\|c(x_k + \alpha p)\|_1 - \mu\|c_k\|_1 \\ &\quad \leqslant \alpha\nabla f_k^{\mathrm{T}} p + \gamma\alpha^2\|p\|^2 + \mu\|c_k + \alpha A_k p\|_1 - \mu\|c_k\|_1, \end{aligned}$$

where $\gamma$ bounds the second-derivative terms in $f$ and $c$. □

# §18.3 Algorithmic Development

## Proof (cont'd).

If $p = p_k$ is given by (6), we have that $A_k p_k = -c_k$, so for $\alpha \leqslant 1$ we have that

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \leqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] + \alpha^2\gamma\|p_k\|^2.$$

By arguing similarly, we also obtain the following lower bound:

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \geqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] - \alpha^2\gamma\|p_k\|^2.$$

Taking limits, we conclude that the directional derivative of $\phi_1$ in the direction $p_k$ is given by

$$D(\phi_1(x_k; \mu); p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\,, \tag{23}$$

which proves (21). The fact that $p_k$ satisfies the first equation in (6) implies that

$$D(\phi_1(x_k; \mu); p_k) = -p_k^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p_k + p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1} - \mu\|c_k\|_1\,. \qquad \square$$

# §18.3 Algorithmic Development

## Proof (cont'd).

If $p = p_k$ is given by (6), we have that $A_k p_k = -c_k$, so for $\alpha \leqslant 1$ we have that

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \leqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] + \alpha^2\gamma\|p_k\|^2.$$

By arguing similarly, we also obtain the following lower bound:

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \geqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] - \alpha^2\gamma\|p_k\|^2.$$

Taking limits, we conclude that the directional derivative of $\phi_1$ in the direction $p_k$ is given by

$$D(\phi_1(x_k; \mu); p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\,, \tag{23}$$

which proves (21). The fact that $p_k$ satisfies the first equation in (6) implies that

$$D(\phi_1(x_k; \mu); p_k) = -p_k^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p_k + p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1} - \mu\|c_k\|_1\,. \qquad \square$$

# §18.3 Algorithmic Development

## Proof (cont'd).

If $p = p_k$ is given by (6), we have that $A_k p_k = -c_k$, so for $\alpha \leqslant 1$ we have that

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \leqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] + \alpha^2\gamma\|p_k\|^2.$$

By arguing similarly, we also obtain the following lower bound:

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \geqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] - \alpha^2\gamma\|p_k\|^2.$$

Taking limits, we conclude that the directional derivative of $\phi_1$ in the direction $p_k$ is given by

$$D(\phi_1(x_k; \mu); p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\,, \tag{23}$$

which proves (21). The fact that $p_k$ satisfies the first equation in (6) implies that

$$D(\phi_1(x_k; \mu); p_k) = -p_k^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p_k + p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1} - \mu\|c_k\|_1\,. \qquad \square$$

# §18.3 Algorithmic Development

## Proof (cont'd).

If $p = p_k$ is given by (6), we have that $A_k p_k = -c_k$, so for $\alpha \leqslant 1$ we have that

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \leqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] + \alpha^2\gamma\|p_k\|^2.$$

By arguing similarly, we also obtain the following lower bound:

$$\phi_1(x_k + \alpha p_k; \mu) - \phi_1(x_k; \mu) \geqslant \alpha\left[\nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\right] - \alpha^2\gamma\|p_k\|^2.$$

Taking limits, we conclude that the directional derivative of $\phi_1$ in the direction $p_k$ is given by

$$D(\phi_1(x_k; \mu); p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1\,, \tag{23}$$

which proves (21). The fact that $p_k$ satisfies the first equation in (6) implies that

$$D(\phi_1(x_k; \mu); p_k) = -p_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p_k + p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1} - \mu\|c_k\|_1\,. \qquad \square$$

# §18.3 Algorithmic Development

## Proof (cont'd).

From the second equation $A_k p_k = -c_k$ in (6), we can replace the term $p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1}$ in

$$D(\phi_1(x_k;\mu);p_k) = -p_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p_k + p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1} - \mu \|c_k\|_1$$

(from the previous page) by $-c_k^{\mathrm{T}} \lambda_{k+1}$. Inequality

$$D(\phi_1(x_k;\mu);p_k) \leqslant -p_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p_k - (\mu - \|\lambda_{k+1}\|_\infty)\|c_k\|_1 \qquad (22)$$

is then concluded by making this substitution in the first identity (in this slide) above and invoking the inequality

$$-c_k^{\mathrm{T}} \lambda_{k+1} \leqslant \|c_k\|_1 \|\lambda_{k+1}\|_\infty .$$

□

# §18.3 Algorithmic Development

## Proof (cont'd).

From the second equation $A_k p_k = -c_k$ in (6), we can replace the term $p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1}$ in

$$D(\phi_1(x_k;\mu);p_k) = -p_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p_k + p_k^{\mathrm{T}} A_k^{\mathrm{T}} \lambda_{k+1} - \mu \|c_k\|_1$$

(from the previous page) by $-c_k^{\mathrm{T}} \lambda_{k+1}$. Inequality

$$D(\phi_1(x_k;\mu);p_k) \leqslant -p_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p_k - (\mu - \|\lambda_{k+1}\|_\infty)\|c_k\|_1 \qquad (22)$$

is then concluded by making this substitution in the first identity (in this slide) above and invoking the inequality

$$-c_k^{\mathrm{T}} \lambda_{k+1} \leqslant \|c_k\|_1 \|\lambda_{k+1}\|_\infty .$$

□

# §18.3 Algorithmic Development

根據 (22) 的結果，如果 $p_k \neq 0$、$\nabla^2_{xx}\mathcal{L}_k$ 是正定的，且

$$\mu > \|\lambda_{k+1}\|_\infty \tag{24}$$

的話，$p_k$ 會是 $\phi_1$ 的下降方向。更詳細的分析顯示 $\nabla^2_{xx}\mathcal{L}_k$ 是正定的這個假設可以被放鬆；我們只需要 reduced-Hessian $Z_k^{\mathrm{T}}\nabla^2_{xx}\mathcal{L}_k Z_k$ 是正定的即可。

在每次迭代中選擇 $\phi_1(x;\mu)$ 中懲罰參數 $\mu$ 的新值的一種直觀策略是讓下一次迭代的 $\mu$ 值滿足 (24)，但留有一定的餘地。然而，這種策略可能會選擇不適當的 $\mu$ 值，並且經常干擾迭代的進展。

# §18.3 Algorithmic Development

另一種基於 (21) 的替代方法是要求方向導數在某種程度上負的夠大，即

$$D(\phi_1(x_k;\mu);p_k) = \nabla f_k^{\mathrm{T}} p_k - \mu\|c_k\|_1 \leqslant -\rho\mu\|c_k\|_1,$$

其中 $\rho \in (0,1)$。上述不等式成立如果我們挑選

$$\mu \geqslant \frac{\nabla f_k^{\mathrm{T}} p_k}{(1-\rho)\|c_k\|_1}. \tag{25}$$

這個選擇不依賴於 Lagrange 乘數，並在實踐中表現良好。

# §18.3 Algorithmic Development

一個適用於 line search 和 trust-region 方法的更有效的選擇 $\mu$ 的策略則考慮了步進量對 merit function 模型的影響。我們通過以下方式定義 $\phi_1$ 的（分段）二次模型：

$$q_\mu(p) = f_k + \nabla f_k^{\mathrm{T}} p + \frac{\sigma}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p + \mu m(p), \tag{26}$$

其中

$$m(p) = \|c_k + A_k p\|_1,$$

且 $\sigma$ 是一個待定的參數。在計算了步進量 $p_k$ 之後，我們選擇足夠大的懲罰參數 $\mu$，使得

$$q_\mu(0) - q_\mu(p_k) \geqslant \rho\mu[m(0) - m(p_k)], \tag{27}$$

其中 $\rho \in (0, 1)$。根據 (26) 和 (5b)，不等式 (27) 成立的條件是

$$\mu \geqslant \frac{\nabla f_k^{\mathrm{T}} p_k + (\sigma/2) p_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p_k}{(1-\rho)\|c_k\|_1}. \tag{28}$$

# §18.3 Algorithmic Development

如果前一次 SQP 方法迭代中 $\mu$ 的值滿足 (28)，則於下次迭代保持不變，否則 $\mu$ 值將增加以使其（在一定餘地下）滿足 (28)。常數 $\sigma$ 則用於處理 Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 非正定的情況。定義 $\sigma$ 為

$$\sigma = \begin{cases} 1 & \text{if } p_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k p_k > 0, \\ 0 & \text{otherwise.} \end{cases}$$

很容易驗證如果 $\mu$ 滿足 (28)，上述 $\sigma$ 的選擇確保了

$$D(\phi_1(x_k;\mu);p_k) \leqslant -\rho\mu\|c_k\|_1,$$

因此 $p_k$ 是 $\phi_1$ 的下降方向。當 $\sigma = 1$ 且 $p_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k p_k < 0$ 時，這個結論不一定成立。通過比較 (25) 和 (28)，我們可以看到：當 $\sigma > 0$ 時，基於 (27) 的策略會選擇一個更大的懲罰參數，從而更加重視限制函數值的減少。此特性是有利的，因為在這種情況下，步進量更有可能被 merit function 接受。

# §18.3 Algorithmic Development

- **Second-order correction**

在第 15 章中，我們通過一示例展示了許多 merit function 可能阻礙優化算法的進展，這種現象被稱為 Maratos 效應。現在我們展示了該示例中分析的步進量實際上是由 SQP 方法產生的。

**Example (Revisit of one example in Chapter 15)**

Consider problem

$$\min f(x_1, x_2) = 2(x_1^2 + x_2^2 - 1) - x_1 \quad \text{subject to} \quad x_1^2 + x_2^2 = 1\,.$$

At the iterate $\boldsymbol{x}_k = (\cos\theta, \sin\theta)^{\mathrm{T}}$, let us compute a search direction $p_k$ by solving the SQP sub-problem

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{5a}$$

subject to

$$A_k p + c_k = 0 \tag{5b}$$

with $\nabla_{xx}^2 \mathcal{L}_k$ replaced by $\nabla_{xx}^2 \mathcal{L}(\boldsymbol{x}_*, \lambda_*) = \mathrm{I}$.

# §18.3 Algorithmic Development

## Example (cont'd)

Since

$$f_k = -\cos\theta, \quad \nabla f_k = \begin{bmatrix} 4\cos\theta - 1 \\ 4\sin\theta \end{bmatrix}, \quad A_k^{\mathrm{T}} = \begin{bmatrix} 2\cos\theta \\ 2\sin\theta \end{bmatrix},$$

the quadratic sub-problem (5) takes the form

$$\min_{p}(4\cos\theta - 1)p_1 + 4\sin\theta p_2 + \frac{1}{2}p_1^2 + \frac{1}{2}p_2^2$$

subject to $p_2 + \cot\theta p_1 = 0$. By solving this sub-problem, we obtain the direction

$$p_k = \begin{bmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{bmatrix},$$

which coincides with $(23)_{15}$.

# §18.3 Algorithmic Development

## Example (cont'd)

Since

$$f_k = -\cos\theta, \quad \nabla f_k = \begin{bmatrix} 4\cos\theta - 1 \\ 4\sin\theta \end{bmatrix}, \quad A_k^{\mathrm{T}} = \begin{bmatrix} 2\cos\theta \\ 2\sin\theta \end{bmatrix},$$

the quadratic sub-problem (5) takes the form

$$\min_{p}(4\cos\theta - 1)p_1 + 4\sin\theta p_2 + \frac{1}{2}p_1^2 + \frac{1}{2}p_2^2$$

subject to $p_2 + \cot\theta p_1 = 0$. By solving this sub-problem, we obtain the direction

$$p_k = \begin{bmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{bmatrix},$$

which coincides with $(23)_{15}$.

# §18.3 Algorithmic Development

在 §15.4 中，我們提到與 Maratos 效應相關的困難可以通過二階修正來克服。有各種應用此一技術的方法；我們接下來描述其中一種可能的實現方式。

假設 SQP 方法從 (8) 中計算出一個步進量 $p_k$。如果這個步進量導致 $\phi_1$ 函數值增加，可能的原因是我們對限制式的線性近似不夠精確。為了克服這個缺陷，我們可以重新解 (8)，其中原先的線性限制 (8b) 與 (8c) 中的限制函數 $c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p$ 被替換為二次近似

$$c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 c_i(x_k) p. \tag{29}$$

然而，即使限制式的 Hessian 矩陣是可用的，得到的二次子問題可能仍然難以解決。因此，我們評估新點 $x_k + p_k$ 在限制函數的函數值，並利用下頁的近似。

# §18.3 Algorithmic Development

根據泰勒定理，我們有

$$c_i(x_k + p_k) \approx c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p_k + \frac{1}{2} p_k^{\mathrm{T}} \nabla^2 c_i(x_k) p_k. \tag{30}$$

假設（仍未知的）二階步進量 $p$ 不會與 $p_k$ 差異太大，我們可以將 (29) 中的最後一項近似為

$$p^{\mathrm{T}} \nabla^2 c_i(x_k) p = p_k^{\mathrm{T}} \nabla^2 c_i(x_k) p_k. \tag{31}$$

通過在 (29) 中進行這個替換，並利用 (30)，我們得到以下的二階修正子問題：

$$\begin{aligned} \min_p \ & \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \\ \text{subject to} \quad & \nabla c_i(x_k)^{\mathrm{T}} p + d_i = 0, i \in \mathcal{E}, \\ & \nabla c_i(x_k)^{\mathrm{T}} p + d_i \geqslant 0, i \in \mathcal{I}, \end{aligned}$$

其中

$$d_i = c_i(x_k + p_k) - \nabla c_i(x_k)^{\mathrm{T}} p_k, \quad i \in \mathcal{E} \cup \mathcal{I}.$$

# §18.3 Algorithmic Development

進行二階修正步進量需要計算 $c_i(x_k + p_k)$（其中 $i \in \mathcal{E} \cup \mathcal{I}$），因此最好不要在每次 merit function 函數值增加時都使用這個修正策略。一種策略是只在 merit function 增加伴隨著限制的範數增加時才使用它。

可以證明，當步進量 $p_k$ 是由 SQP 方法 (8) 生成時，接近滿足二階充分條件的解時，上述算法會採取完整步進量 $p_k$ 或修正步進量 $p_k + \hat{p}_k$（$\hat{p}_k = p - p_k$）。Merit function 不會干擾迭代，因此實現了超線性收斂，就像局部算法一樣。

# §18.4 A Practical Line Search SQP Method

根據前一節的討論，我們可以看到有許多不同的 line search SQP 方法，它們在 Hessian 矩陣近似的計算方式、步進量接受機制以及其它演算法特徵方面存在差異。我們現在將一些這些想法融入一個具體的、實用的 SQP 算法中，用於解決非線性規劃問題

$$\min f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geqslant 0, i \in \mathcal{I}. \end{cases} \tag{7}$$

為了保持描述的簡單性，我們不會使用包括像

$$\min_{x,v,w,t} f(x) + \mu \sum_{i \in \mathcal{E}} (v_i + w_i) + \mu \sum_{i \in \mathcal{I}} t_i \tag{9a}$$

$$\text{subject to} \quad c_i(x) = v_i - w_i, i \in \mathcal{E}, \tag{9b}$$

$$c_i(x) \geqslant -t_i, i \in \mathcal{I}, \tag{9c}$$

$$v, w, t \geqslant 0, \tag{9d}$$

這樣的機制來確保子問題的可行性，或者採取二階修正步進量。

# §18.4 A Practical Line Search SQP Method

相反，搜索方向僅通過解決子問題

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p \tag{8a}$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{8b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}, \tag{8c}$$

獲得。我們還假設二次規劃 (8) 是凸的，因此我們可以通過在第 16 章中描述的二次規劃的 active set 方法 (Algorithm 16.3) 來解決它。

# §18.4 A Practical Line Search SQP Method

**Algorithm 18.3** (Line Search SQP Algorithm).

Choose parameters $\eta \in (0, 0.5)$, $\tau \in (0, 1)$, and an initial pair $(x_0, \lambda_0)$;

Evaluate $f_0$, $\nabla f_0$, $c_0$, $A_0$;

If a quasi-Newton approximation is used, choose an initial $n \times n$ symmetric positive definite Hessian approximation $B_0$, otherwise compute $\nabla^2_{xx}\mathcal{L}_0$;

**repeat** until a convergence test is satisfied

Compute $p_k$ by solving (8); let $\widehat{\lambda}$ be the corresponding Lagrange multiplier;

Set $p_\lambda \leftarrow \widehat{\lambda} - \lambda_k$;

Choose $\mu_k$ to satisfy (28) with $\sigma = 1$;

Set $\alpha_k \leftarrow 1$;

**while** $\phi_1(x_k + \alpha_k p_k; \mu_k) > \phi_1(x_k; \mu_k) + \eta\alpha_k D_1(\varphi(x_k; \mu_k)p_k)$

Reset $\alpha_k \leftarrow \tau_\alpha \alpha_k$ for some $\tau_\alpha \in (0, \tau]$;

**end** (while)

# §18.4 A Practical Line Search SQP Method

**while** $\phi_1(x_k + \alpha_k p_k; \mu_k) > \phi_1(x_k; \mu_k) + \eta\alpha_k D_1(\varphi(x_k; \mu_k)p_k)$
  Reset $\alpha_k \leftarrow \tau_\alpha \alpha_k$ for some $\tau_\alpha \in (0, \tau]$;
**end** (while)
Set $x_{k+1} \leftarrow x_k + \alpha_k p_k$ and $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k p_\lambda$;
Evaluate $f_{k+1}$, $\nabla f_{k+1}$, $c_{k+1}$, $A_{k+1}$, (and possibly $\nabla^2_{xx}\mathcal{L}_{k+1}$);
If a quasi-Newton approximation is used, set $s_k \leftarrow \alpha_k p_k$ and
  $y_k \leftarrow \nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})$, and obtain $B_{k+1}$
  by updating $B_k$ using a quasi-Newton formula;
**end** (repeat)

# §18.4 A Practical Line Search SQP Method

**Algorithm 18.3** (Line Search SQP Algorithm).

Choose $\eta \in (0, 0.5)$, $\tau \in (0, 1)$, and an initial pair $(x_0, \lambda_0)$;
Evaluate $f_0$, $\nabla f_0$, $c_0$, $A_0$;
If a quasi-Newton approximation is used, choose an $n \times n$ positive definite Hessian approximation $B_0$, otherwise compute $\nabla^2_{xx}\mathcal{L}_0$;
**repeat** until a convergence test is satisfied
  Compute $p_k$ by solving (8); let $\widehat{\lambda}$ be the Lagrange multiplier;
  Set $p_\lambda \leftarrow \widehat{\lambda} - \lambda_k$ and $\alpha_k \leftarrow 1$;
  Choose $\mu_k$ to satisfy (28) with $\sigma = 1$;
  **while** $\phi_1(x_k + \alpha_k p_k; \mu_k) > \phi_1(x_k; \mu_k) + \eta\alpha_k D_1(\varphi(x_k; \mu_k)p_k)$
    Reset $\alpha_k \leftarrow \tau_\alpha \alpha_k$ for some $\tau_\alpha \in (0, \tau]$;
  **end** (while)
  Set $x_{k+1} \leftarrow x_k + \alpha_k p_k$ and $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k p_\lambda$;
  Evaluate $f_{k+1}$, $\nabla f_{k+1}$, $c_{k+1}$, $A_{k+1}$, (and possibly $\nabla^2_{xx}\mathcal{L}_{k+1}$);
  If a quasi-Newton approximation is used, set $s_k \leftarrow \alpha_k p_k$ and $y_k \leftarrow \nabla_x\mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x\mathcal{L}(x_k, \lambda_{k+1})$, and obtain $B_{k+1}$ by updating $B_k$ using a quasi-Newton formula;
**end** (repeat)

# §18.4 A Practical Line Search SQP Method

通過熱啟動程序，我們可以在解決二次子問題時實現顯著的節省。例如，我們可以將每個二次規劃子問題的工作集初始化為前一次 SQP 迭代的最終 active set。

在 Algorithm 18.3 中，我們沒有給出 quasi-Newton 近似的細節。我們可以使用，例如，適用於大規模問題的有限記憶 BFGS 方法。如果我們使用 exact 的 Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$，我們假設它根據需要修改為在等式限制的 null space 上是正定的。

我們可以在內部的 "while" 迴圈中使用一個 filter（參見 §15.4），而不是使用 merit function 來確定步長 $\alpha_k$。如 §15.4 所討論的，如果由 backtracking line search 生成的試驗步長小於一個給定的閾值 (threshold)，則會調用一個 feasibility restoration phase。無論是使用 merit function 還是 filter，都可以將二階修正等機制納入其中，以克服 Maratos 效應。

# §18.5 Trust-Region SQP Methods

Trust-region SQP 方法具有幾個吸引人的特性。其中包括它們不需要在 (8) 中的海森矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 是正定的，即使在存在 Hessian 矩陣和 Jacobian 矩陣 singularities 的情況下，它們也能控制步進量的質量，並提供強制全域收斂的機制。一些實現遵循 IQP 方法並解決一個不等式限制的子問題，而其它則遵循 EQP 方法。

最簡單的制定 trust-region SQP 方法的方式是將 trust-region 限制添加到子問題 (8) 中，如下所示：

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k p \tag{32a}$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{32b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}, \tag{32c}$$

$$\|p\| \leqslant \Delta_k. \tag{32d}$$

# §18.5 Trust-Region SQP Methods

Trust-region SQP 方法具有幾個吸引人的特性。其中包括它們不需要在 (8) 中的海森矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 是正定的，即使在存在 Hessian 矩陣和 Jacobian 矩陣 singularities 的情況下，它們也能控制步進量的質量，並提供強制全域收斂的機制。一些實現遵循 IQP 方法並解決一個不等式限制的子問題，而其它則遵循 EQP 方法。

最簡單的制定 trust-region SQP 方法的方式是將 trust-region 限制添加到子問題 (8) 中，如下所示：

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k p \tag{32a}$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{32b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}, \tag{32c}$$

$$\|p\| \leqslant \Delta_k. \tag{32d}$$

# §18.5 Trust-Region SQP Methods

即使限制 (32b)、(32c) 是相容的，由於 trust-region 限制 (32d) 的緣故，該子問題可能並非總是有解的。我們在圖 1 中展示了僅包含一個等式限制的情況。該限制之線性化由實線表示，而 trust region 由半徑 $\Delta_k$ 的圓圈表示，此例中任何滿足線性化限制的步進量 $p$ 位於 trust region 之外。由此例中可以看出，若限制解的範數，相容的等式和不等式系統可能無解。

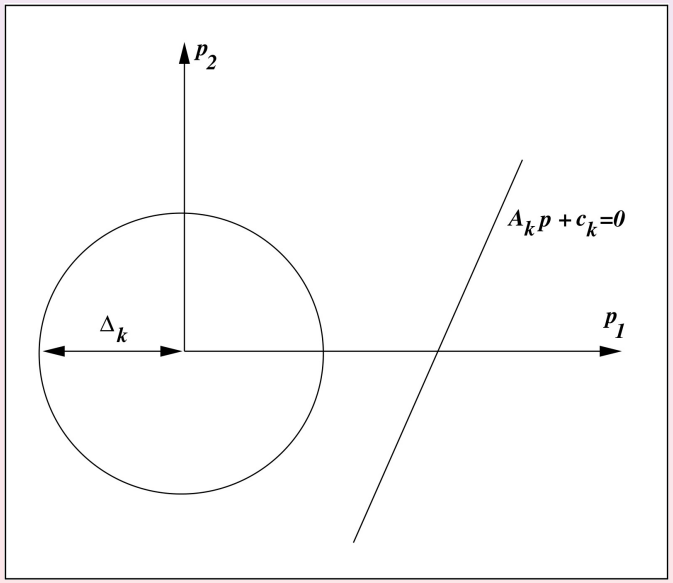


Figure 1: Inconsistent constraints in trust-region model.

# §18.5 Trust-Region SQP Methods

為了解決線性限制 (32b)、(32c) 和 trust-region 限制 (32d) 之間可能的衝突，單純地增加 $\Delta_k$ 直到滿足線性限制的步進量 $p$ 與 trust region 相交是不合適的。這種方法會背離使用 trust region 的初衷，因為 trust region 是用來定義一個我們信任模型 (32a)-(32c) 準確反映目標和限制函數行為的區域。從分析上來看，這會損害演算法的收斂性質。

更適當的觀點是，在每一步都精確滿足線性化限制的要求並無必要；相反，我們應該在每一步中努力改善這些限制的可行性，只有在 trust region 限制允許的情況下才完全滿足它們。這種觀點是本節討論的三種方法 relaxation 法、penalty 法和 filter 法的基礎。

# §18.5 Trust-Region SQP Methods

為了解決線性限制 (32b)、(32c) 和 trust-region 限制 (32d) 之間可能的衝突，單純地增加 $\Delta_k$ 直到滿足線性限制的步進量 $p$ 與 trust region 相交是不合適的。這種方法會背離使用 trust region 的初衷，因為 trust region 是用來定義一個我們信任模型 (32a)-(32c) 準確反映目標和限制函數行為的區域。從分析上來看，這會損害演算法的收斂性質。

更適當的觀點是，在每一步都精確滿足線性化限制的要求並無必要；相反，我們應該在每一步中努力改善這些限制的可行性，只有在 trust region 限制允許的情況下才完全滿足它們。這種觀點是本節討論的三種方法 relaxation 法、penalty 法和 filter 法的基礎。

# §18.5 Trust-Region SQP Methods

- **A relaxation method for equality-constrained optimization**

我們將在只具等式限制的優化問題 (1) 的背景下描述這種方法。針對一般的非線性規劃問題的 relaxation 法，因其使用了內點法技術的緣故，我們將延至第 19 章才說明。在迭代 $x_k$ 中，我們通過解決子問題來計算 SQP 步驟：

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \tag{33a}$$

subject to

$$A_k p + c_k = r_k, \tag{33b}$$

$$\|p\|_2 \leqslant \Delta_k. \tag{33c}$$

# §18.5 Trust-Region SQP Methods

選擇放寬向量 $r_k$ 需要仔細考慮，因為它會影響方法的效率。我們的目標是選擇 $r_k$ 為使得對於某個縮小的 trust region 半徑 $\Delta_k$ 限制 (33b)、(33c) 是相容的最小的向量。為了做到這一點，我們首先解決下面的子問題：

$$\min_v \|A_k v + c_k\|_2^2 \tag{35a}$$

$$\text{subject to}$$

$$\|v\|_2 \leqslant 0.8\Delta_k. \tag{35b}$$

將這個子問題的解記為 $v_k$，並定義

$$r_k = A_k v_k + c_k. \tag{34}$$

接下來，我們通過解決 (33) 來計算步進量 $p_k$，定義新的迭代點 $x_{k+1} = x_k + p_k$ 並使用最小平方公式 (15) 獲得新的乘子估計 $\lambda_{k+1}$。注意，限制 (33b)、(33c) 是相容的，因為至少有向量 $p = v_k$ 同時滿足限制。

# §18.5 Trust-Region SQP Methods

乍看之下，這種方法似乎不切實際，因為問題 (33) 和 (35) 並不容易解決，特別是當 $\nabla^2_{xx}\mathcal{L}_k$ 是不定 (indefinite) 時。幸運的是，我們可以設計出高效的程序來計算這些問題的有用的不精確解。

我們通過在第 4 章中描述的 dogleg 法來解決輔助子問題 (35)。該方法需要一個 Cauchy step $p^U$ 和一個 "Newton step" $p^B$，它們是優化問題 (35a) 分別在方向 $-A_k^{\mathrm{T}}c_k$ 上的 minimizer 以及在無限制條件下的 minimizer。由於 (35a) 中的 Hessian 矩陣是不可逆的，存在無窮多種可能的 $p^B$ 選擇，所有這些選擇都滿足 $A_kp^B + c_k = 0$。我們通過設置 $p^B = -A_k^{\mathrm{T}}[A_kA_k^{\mathrm{T}}]^{-1}c_k$ 來選擇具有最小範數的 $p^B$。

# §18.5 Trust-Region SQP Methods

現在我們將 $v_k$ 取為沿著 $p^U$、$p^B$ 和公式

$$\tilde{p}(\tau) = \begin{cases} \tau p^U & \text{if } 0 \leqslant \tau \leqslant 1, \\ p^U + (\tau - 1)(p^B - p^U) & \text{if } 1 \leqslant \tau \leqslant 2. \end{cases} \qquad (14)_4$$

定義的路徑上 (35a) 的 minimizer。

計算 (33) 的近似解 $p_k$ 的首選技術是 Algorithm 16.2 中的投影共軛梯度法。我們將此算法應用於只具等式限制二次規劃 (33a)–(33b)，監視 trust region 限制 (33c) 的滿足情況，並在達到該區域邊界或檢測到負曲率時停止；請參見 §7.1。Algorithm 16.2 需要一個可行的起始點，可以選擇為 $v_k$。

# §18.5 Trust-Region SQP Methods

現在我們將 $v_k$ 取為沿著 $p^U$、$p^B$ 和公式

$$\tilde{p}(\tau) = \begin{cases} \tau p^U & \text{if } 0 \leqslant \tau \leqslant 1, \\ p^U + (\tau - 1)(p^B - p^U) & \text{if } 1 \leqslant \tau \leqslant 2. \end{cases} \qquad (14)_4$$

定義的路徑上 (35a) 的 minimizer。

計算 (33) 的近似解 $p_k$ 的首選技術是 Algorithm 16.2 中的投影共軛梯度法。我們將此算法應用於只具等式限制二次規劃 (33a)–(33b)，監視 trust region 限制 (33c) 的滿足情況，並在達到該區域邊界或檢測到負曲率時停止；請參見 §7.1。Algorithm 16.2 需要一個可行的起始點，可以選擇為 $v_k$。

# §18.5 Trust-Region SQP Methods

這種方法中很適合的 merit function 是非光滑的 $\ell_2$ 函數 $\phi_2(x;\mu) = f(x) + \mu\|c(x)\|_2$。我們通過以下函數來模擬它：

$$q_\mu(p) = f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p + \mu m(p),$$

其中

$$m(p) = \|c_k + A_k p\|_2,$$

(見 (26))。我們選擇足夠大的懲罰參數，使不等式 (27) 得到滿足。為了判斷步進量 $p_k$ 的可接受性，我們監控

$$\rho_k = \frac{\mathrm{ared}_k}{\mathrm{pred}_k} = \frac{\phi_2(x_k,\mu) - \phi_2(x_k + p_k,\mu)}{q_\mu(0) - q_\mu(p_k)}. \tag{36}$$

在下頁我們對只具等式限制的優化問題 (1) 的這種 trust-region SQP 方法進行描述。

# §18.5 Trust-Region SQP Methods

**Algorithm 18.4** (Byrd–Omojokun Trust-Region SQP Method).

Choose constants $\varepsilon > 0$ and $\eta$, $\gamma \in (0, 1)$;
Choose starting point $x_0$, initial trust region $\Delta_0 > 0$;
**for** $k = 0, 1, 2, \cdots$
  Compute $f_k$, $c_k$, $\nabla f_k$, $A_k$;
  Compute multiplier estimates $\widehat{\lambda}_k$ by (15);
  **if** $\|\nabla f_k - A_k^{\mathrm{T}} \widehat{\lambda} k\|_\infty < \varepsilon$ and $\|c_k\|_\infty < \varepsilon$
    **stop** with approximate solution $x_k$;
  Solve normal sub-problem (35) for $v_k$ and compute $r_k$ from (34);
  Compute $\nabla^2_{xx} \mathcal{L}_k$ or a quasi-Newton approximation;
  Compute $p_k$ by applying the projected CG method to (33);
  Choose $\mu_k$ to satisfy (27);
  Compute $\rho_k = \mathrm{ared}_k / \mathrm{pred}_k$;
  **if** $\rho_k > \eta$
    Set $x_{k+1} = x_k + p_k$;

# §18.5 Trust-Region SQP Methods

Choose $\mu_k$ to satisfy (27);
Compute $\rho_k = \text{ared}_k / \text{pred}_k$;
**if** $\rho_k > \eta$
  Set $x_{k+1} = x_k + p_k$;
  Choose $\Delta_{k+1}$ to satisfy $\Delta_{k+1} \geqslant \Delta_k$;
**else**
  Set $x_{k+1} = x_k$;
  Choose $\Delta_{k+1}$ to satisfy $\Delta_{k+1} \leqslant \gamma \|p_k\|$;
**end** (for).

為了避免 Maratos 效應，可以添加二階修正。除了目標函數 $f$ 和限制函數 $c$ 取值的成本之外，該演算法的主要成本在於投影共軛梯度迭代，它需要計算 Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 與向量的乘積，以及使用投影矩陣 (16.32) 進行矩陣分解和回代的成本；詳見 §16.3。

# §18.5 Trust-Region SQP Methods

- **S$\ell_1$QP (Sequential $\ell_1$ Quadratic Programming)**

在這種方法中，我們將線性化的限制條件

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = 0, i \in \mathcal{E}, \tag{32b}$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant 0, i \in \mathcal{I}, \tag{32c}$$

以 $\ell_1$ 懲罰項的形式移入二次規劃的目標中而得到以下子問題：

$$\begin{aligned} \min_p q_\mu(p) \equiv\ & f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p \\ & + \mu \sum_{i \in \mathcal{E}} \left| c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \right| \\ & + \mu \sum_{i \in \mathcal{I}} \left[ c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \right]^- \\ \text{subject to } & \|p\|_\infty \leqslant \Delta_k, \end{aligned} \tag{37}$$

其中 $\mu$ 是懲罰參數，而符號 $[y]^- = \max\{0, -y\}$ 代表 $y$ 的負部。

# §18.5 Trust-Region SQP Methods

引入鬆弛變量 $v$、$w$、$t$ 後，我們可以重新表述這個問題如下：

$$\min_{p,v,w,t} f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p + \mu \sum_{i \in \mathcal{E}} (v_i + w_i) + \mu \sum_{i \in \mathcal{I}} t_i \quad (38a)$$

subject to

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) = v_i - w_i, \quad i \in \mathcal{E}, \quad (38b)$$

$$\nabla c_i(x_k)^{\mathrm{T}} p + c_i(x_k) \geqslant -t_i, \quad i \in \mathcal{I}, \quad (38c)$$

$$v, w, t \geqslant 0, \quad (38d)$$

$$\|p\|_\infty \leqslant \Delta_k. \quad (38e)$$

這種表述僅僅是 (9) 式的線性化，並同時添加了一個信賴區域限制。

# §18.5 Trust-Region SQP Methods

這個問題的限制條件始終是相容的。由於信賴區域是使用 $\ell_\infty$ 範數定義的，(38) 是一個光滑的二次規劃問題，可以通過二次規劃算法解決。熱啟動策略可以顯著降低 (38) 的解決時間，在實際實施中總是被使用。

自然地，我們可以使用以下的 $\ell_1$ merit function 來確定步進量接受與否：

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^- \tag{39}$$

事實上，(37) 中定義的函數 $q_\mu$ 可以被視為在 $x_k$ 處對 $\phi_1(x,\mu)$ 的模型，在該模型中，我們通過線性化來近似每個限制函數 $c_i$，並將 $f$ 替換為一個包含來自目標和限制的訊息的二次函數。

# §18.5 Trust-Region SQP Methods

在從 (38) 計算步進量 $p_k$ 後，我們通過利用 merit function $\phi_1$ 和 $q_\mu$ 的定義並使用

$$\rho_k = \frac{\text{ared}_k}{\text{pred}_k} = \frac{\phi_1(x_k, \mu) - \phi_1(x_k + p_k, \mu)}{q_\mu(0) - q_\mu(p_k)} \tag{36}$$

來確定比率 $\rho_k$，再根據標準 trust region 規則，亦即在 Algorithm 18.4 中實現的方式，來決定是否接受或拒絕該步進量。為了防止 Maratos 效應的發生，可以添加二階修正步驟。

S$\ell_1$QP 方法具有幾個吸引人的特點：不僅公式 (37) 克服了線性化限制可能的不相容性，還確保 trust-region 限制始終可以滿足。此外，矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 以在子問題 (38) 中不加修改地使用，或者可以用 quasi-Newton 逼近代替。它無需具有正定性要求。

# §18.5 Trust-Region SQP Methods

選擇懲罰參數 $\mu$ 在這種方法的效率中扮演著重要角色。與上述的 SQP 方法不同，該方法僅用懲罰函數來確定試驗點的可接受性，S$\ell_1$QP 算法的步進量 $p_k$ 取決於 $\mu$。$\mu$ 值過小可能使算法偏離解，而過大的 $\mu$ 值可能導致進展緩慢。為了在多種應用場景中獲得良好的實際性能，必須在每次迭代中仔細選擇 $\mu$ 的值；參見下面的 Algorithm 18.5。

# §18.5 Trust-Region SQP Methods

- **Sequential linear quadratic programming (SLQP)**

上述討論的 SQP 方法在每次迭代中需要解決一個一般的（只具不等式限制的）二次問題。解決這個子問題的計算量對於實際可解問題的大小有所限制。此外，在 SQP 方法中融入（不定）二階導數訊息已被證明是困難的。

連續線性二次規劃 (SLQP) 方法試圖通過兩個階段的計算來解決這些問題，而其中解決每個階段中的子問題之計算量大致正比於變量的數量。首先，解決線性規劃 (LP) 以確定一個工作集 $\mathcal{W}$。其次為一個只具等式限制的二次規劃 (EQP) 階段，將前步驟確定下的工作集 $\mathcal{W}$ 中的限制作為等式限制施加在原問題上。該算法的總步進量是線性規劃和只具等式限制階段獲得的步進量的組合，我們現在進行討論。

# §18.5 Trust-Region SQP Methods

在線性規劃 (LP) 階段，我們希望解決以下問題：

$$\min_p f_k + \nabla f_k^{\mathrm{T}} p \tag{40a}$$

subject to

$$c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p = 0, i \in \mathcal{E}, \tag{40b}$$

$$c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \geqslant 0, i \in \mathcal{I}, \tag{40c}$$

$$\|p\|_\infty \leqslant \Delta_k^{\mathrm{LP}}, \tag{40d}$$

這與標準的 SQP 子問題 (32) 的不同之處在於，目標函數中的二階項被省略了，並且使用了 $\ell_\infty$ 範數來定義信賴區域。

# §18.5 Trust-Region SQP Methods

由於問題 (40) 的限制可能不一致，我們改為求其 $\ell_1$ 懲罰形式，其定義如下：

$$\begin{aligned}\min_p \ell_\mu(p) \equiv f_k + \nabla f_k^{\mathrm{T}} p + \mu \sum_{i\in\mathcal{E}} \left| c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \right| \\ + \mu \sum_{i\in\mathcal{I}} \left[ c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \right]^- \qquad (41a)\end{aligned}$$

$$\text{subject to} \quad \|p\|_\infty \leqslant \Delta_k^{\mathrm{LP}}. \qquad (41b)$$

通過引入與問題 (38) 中相同的鬆弛變量，我們可以將問題 (41) 重構為線性規劃 (LP)。公式 (41) 的解，我們記作 $p^{\mathrm{LP}}$，是用第 13 章中的 simplex 法計算得到的。從這個解中，我們得到以下對最佳 active set 的顯式估計：

$$\begin{aligned}\mathcal{A}_k(p^{\mathrm{LP}}) = \left\{ i \in \mathcal{E} \,\middle|\, c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p^{\mathrm{LP}} = 0 \right\} \\ \cup \left\{ i \in \mathcal{I} \,\middle|\, c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p^{\mathrm{LP}} = 0 \right\}.\end{aligned}$$

# §18.5 Trust-Region SQP Methods

同樣地，我們定義違約限制集合 $\mathcal{V}_k$ 為：

$$\mathcal{V}_k(p^{\mathrm{LP}}) = \left\{ i \in \mathcal{E} \,\middle|\, c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p^{\mathrm{LP}} \neq 0 \right\} \cup \left\{ i \in \mathcal{I} \,\middle|\, c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p^{\mathrm{LP}} < 0 \right\}.$$

我們將工作集 $\mathcal{W}_k$ 定義為 active set $\mathcal{A}_k(p^{\mathrm{LP}})$ 的某個線性獨立子集。

為了確保算法在懲罰函數 $\phi_1$ 上取得進展，我們定義柯西步 (Cauchy step) 為：

$$p^C = \alpha^{\mathrm{LP}} p^{\mathrm{LP}}, \tag{42}$$

其中 $\alpha^{\mathrm{LP}} \in (0, 1]$ 是提供在模型 $q_\mu$ 中（見問題 (37)）充分減少的步長。

# §18.5 Trust-Region SQP Methods

給定工作集 $\mathcal{W}_k$，我們現在解決一個只具等式限制的二次規劃問題 (EQP)，將 $\mathcal{W}_k$ 中的限制視為等式，忽略所有其他限制。我們由此得到子問題：

$$\min_p f_k + \frac{1}{2} p^{\mathrm{T}} \nabla^2_{xx} \mathcal{L}_k p + \left( \nabla f_k + \mu_k \sum_{i \in \mathcal{V}_k} \gamma_i \nabla c_i(x_k) \right)^{\mathrm{T}} p \tag{43a}$$

$$\text{subject to} \quad c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p = 0, i \in \mathcal{E}\ Wk, \tag{43b}$$

$$c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p = 0, i \in \mathcal{I} \cap \mathcal{W}_k, \tag{43c}$$

$$\|p\|_2 \leqslant \Delta_k, \tag{43d}$$

其中 $\gamma_i$ 是第 $i$ 個違反限制的代數符號。注意，信賴區域 (43d) 是球形的，且 $\Delta_k$ 與在 (41b) 中使用的信賴區域半徑 $\Delta_k^{\mathrm{LP}}$ 是不同的。

# §18.5 Trust-Region SQP Methods

問題 (43) 通過應用 Algorithm 16.2 的投影共軛梯度法求解向量 $p^Q$，並使用 Steihaug 策略 (Algorithm 7.2) 來處理信賴區域限制。SLQP 方法的總步長 $p_k$ 定義為：

$$p_k = p^C + \alpha^Q(p^Q - p^C),$$

其中 $\alpha^Q \in [0,1]$ 是在模型 $q_\mu$（見問題 (37)）中近似最小化的步長。

# §18.5 Trust-Region SQP Methods

EQP 階段的信賴區域半徑 $\Delta_k$ 使用標準的信賴區域更新策略進行更新。對於 LP 階段，半徑 $\Delta_{k+1}^{\text{LP}}$ 的選擇更為謹慎，因為它會影響我們對最佳 active set 的猜測。$\Delta_{k+1}^{\text{LP}}$ 的值應設置為比總步長 $p_k$ 略大，但需符合其他一些限制 [49]。Hessian $\nabla_{xx}^2 \mathcal{L}_k$ 中使用的乘數估計 $\lambda_k$ 是使用工作集 $\mathcal{W}_k$ 的最小二乘估計（公式 15），並修改使得在 $\mathcal{I}$ 中的 index $i$ 都有 $\lambda_i \geqslant 0$。

SLQP 算法的一個吸引人的特點是，現有技術可以輕鬆解決 LP 和 EQP 子問題的大規模版本。高質量的 LP 套件能夠解決具有大量變量和限制的問題，而 EQP 子問題的解可以通過投影共軛梯度法高效地完成。

# §18.5 Trust-Region SQP Methods

- **A technique for updating the penalty parameter**

我們提到懲罰方法如 S$\ell_1$QP 和 SLQP 對懲罰參數 $\mu$ 的選擇敏感。現在我們討論一個在實踐中被證明有效並且受到全局收斂保證支持的 $\mu$ 選擇程序。目標是選擇足夠小的 $\mu$，以避免在 merit function 中出現不必要的不平衡，但又足夠大，以使步進量在每次迭代中在線性化可行性上取得足夠進展。我們在 S$\ell_1$QP 方法的上下文中介紹這個過程，然後描述其擴展到 SLQP 方法的方式。

# §18.5 Trust-Region SQP Methods

我們通過以下方式定義了在點 $x_k$ 處的限制違反的分段線性模型：

$$m_k(p) = \sum_{i\in\mathcal{E}} \left|c_i(x_k)+\nabla c_i(x_k)^{\mathrm{T}}p\right| + \sum_{i\in\mathcal{I}} \left[c_i(x_k)+\nabla c_i(x_k)^{\mathrm{T}}p)\right]^{-}, \quad (44)$$

這樣 SQP 子問題 (37) 的目標函數可以寫為

$$q_\mu(p) = f_k + \nabla f_k^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}\nabla^2_{xx}\mathcal{L}_k p + \mu m_k(p). \quad (45)$$

我們首先使用先前的懲罰參數值 $\mu_{k-1}$，解決二次規劃子問題 (37)（或等價的 (38)）。如果滿足限制式 (38b)、(38c)，且鬆弛變量 $v_i$、$w_i$、$t_i$ 全部等於零（即 $m_k(p_k) = 0$），則當前的 $\mu$ 值是適當的，我們設置 $\mu_k = \mu_{k-1}$。這是一種幸運的情況，我們可以通過步進量 $p_k$ 實現線性化的可行性，且該步進量的範數不大於 trust region 半徑。

# §18.5 Trust-Region SQP Methods

如果 $m_k(p) > 0$，則增加懲罰參數可能是合適的。問題是：要增加多少？為了獲得一個參考值，我們使用“無限”值的 $\mu$ 重新解決 QP

$$\begin{aligned}\min_p q_\mu(p) \equiv f_k + \nabla f_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k p \\ + \mu \sum_{i \in \mathcal{E}} \left| c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \right| \\ + \mu \sum_{i \in \mathcal{I}} \left[ c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} p \right]^- \\ \text{subject to } \|p\|_\infty \leqslant \Delta_k.\end{aligned} \tag{37}$$

這意味著在 (37) 中，目標函數被 $m_k(p)$ 取代。在計算出新步進量 $p^\infty$ 後，我們有兩種可能的結果。如果 $m_k(p^\infty) = 0$，表示在信任區域內線性化限制是可行的，我們選擇 $\mu_k > \mu_{k-1}$，使得 $m_k(p_k) = 0$。否則，如果 $m_k(p^\infty) > 0$，我們選擇 $\mu_k \geqslant \mu_{k-1}$，使得 $p_k$ 造成的 $m_k$ 減少至少是 $p^\infty$ 給出的（最優）減少的一部分。

# §18.5 Trust-Region SQP Methods

在所有情況下，選擇 $\mu_k > \mu_{k-1}$ 是通過逐步增加 $\mu$ 的當前數值（例如說增加 10 倍），並重新解二次規劃問題 (37) 來實現的。為了更精確地描述這個策略，我們將二次規劃問題 (37) 的解寫為 $p(\mu)$，以強調其對應懲罰參數的依賴性。同樣的，$p^\infty$ 表示在 trust-region 限制 (38e) 下對 $m_k(p)$ 進行最小化的解。以下算法描述了懲罰參數 $\mu_k$ 的選擇以及計算 S$\ell_1$QP 步進量 $p_k$ 的過程。

# §18.5 Trust-Region SQP Methods

**Algorithm 18.5** (Penalty Update and Step Computation).
Initial data: $x_k$, $\mu_{k-1} > 0$, $\Delta_k > 0$, and parameters $\varepsilon_1$, $\varepsilon_2 \in (0, 1)$.
Solve the sub-problem (38) with $\mu = \mu_{k-1}$ to obtain $p(\mu_{k-1})$;
**if** $m_k(p(\mu_{k-1})) = 0$
  Set $\mu^+ \leftarrow \mu_{k-1}$;
**else**
  Compute $p^\infty$;
  **if** $m_k(p^\infty) = 0$
    Find $\mu^+ > \mu_{k-1}$ such that $m_k(p(\mu^+)) = 0$;
  **else**
    Find $\mu^+ \geqslant \mu_{k-1}$ such that $m_k(0) - m_k(p(\mu^+)) \geqslant \varepsilon_1[m_k(0) - m_k(p^\infty)]$;
  **end** (if)
**end** (if)
Increase $\mu^+$ if necessary to satisfy

$$q_{\mu^+}(0) - q_{\mu^+}(p(\mu^+)) \geqslant \varepsilon_2\mu^+[m_k(0) - m_k(p(\mu^+))]; \quad (27')$$

Set $\mu_k \leftarrow \mu^+$ and $p_k \leftarrow p(\mu^+)$.

# §18.5 Trust-Region SQP Methods

儘管 Algorithm 18.5 需要解決一些額外的二次規劃問題，但我們希望通過更快地識別適當的懲罰參數值，來減少總迭代次數（和總二次規劃求解次數），而不像基於可行性監控的規則那樣（參見 Framework 17.2）。

數值經驗表明，在使用 SLQP 方法時，當對 Algorithm 18.5 進行適應 (adaptation) 時，可以實現上述更快找到 $\mu$ 與減少總巡代次數的這些目標。這種適應是通過在 $q_\mu$ 的定義 (37) 中設置 $\nabla^2_{xx}\mathcal{L}_k = 0$，並應用 Algorithm 18.5 來確定 $\mu$ 並計算 LP 步進量 $p^{\text{LP}}$ 而獲得的。在這種情況下，Algorithm 18.5 需要額外的 LP 求解通常是計算量不高的，需要相對較少的單純形迭代，因為我們可以利用先前解決的 LP 的熱啟動信息，使用不同的懲罰參數值。

# §18.5 Trust-Region SQP Methods

儘管 Algorithm 18.5 需要解決一些額外的二次規劃問題，但我們希望通過更快地識別適當的懲罰參數值，來減少總迭代次數（和總二次規劃求解次數），而不像基於可行性監控的規則那樣（參見 Framework 17.2）。

數值經驗表明，在使用 SLQP 方法時，當對 Algorithm 18.5 進行適應 (adaptation) 時，可以實現上述更快找到 $\mu$ 與減少總巡代次數的這些目標。這種適應是通過在 $q_\mu$ 的定義 (37) 中設置 $\nabla^2_{xx}\mathcal{L}_k = 0$，並應用 Algorithm 18.5 來確定 $\mu$ 並計算 LP 步進量 $p^{\mathrm{LP}}$ 而獲得的。在這種情況下，Algorithm 18.5 需要額外的 LP 求解通常是計算量不高的，需要相對較少的單純形迭代，因為我們可以利用先前解決的 LP 的熱啟動信息，使用不同的懲罰參數值。

# §18.6 Nonlinear Gradient Projection

在 §16.7 中，我們討論了界限限制二次規劃的梯度投影法。將此方法擴展到以下問題並不困難：

$$\min f(x) \quad \text{subject to} \quad \ell \leqslant x \leqslant u, \tag{46}$$

其中 $f$ 是非線性函數，$\ell$ 和 $u$ 分別是下界和上界的向量。

我們首先描述一種線搜索方法。在當前迭代點 $x_k$ 我們構建二次模型：

$$q_k(x) = f_k + \nabla f_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} B_k (x - x_k), \tag{47}$$

其中 $B_k$ 是 $\nabla^2 f(x_k)$ 的正定近似 (positive definite approximation)。接下來，我們使用二次規劃的梯度投影法 (Algorithm 16.5) 來尋找子問題

$$\min q_k(x) \quad \text{subject to} \quad \ell \leqslant x \leqslant u \tag{48}$$

的近似解 $\hat{x}$。

# §18.6 Nonlinear Gradient Projection

搜索方向定義為 $p_k = \hat{x} - x_k$，新的迭代點為 $x_{k+1} = x_k + \alpha_k p_k$，其中步長 $\alpha_k$ 選擇滿足條件：

$$f(x_k + \alpha_k p_k) \leqslant f(x_k) + \eta \alpha_k \nabla f_k^{\mathrm{T}} p_k,$$

其中參數 $\eta \in (0, 1)$。

要證明搜索方向 $p_k$ 確實是目標函數的下降方向，我們利用 Algorithm 16.5 的性質，如 §16.7 所討論的。回想一下，該方法沿著分段線性路徑（投影最陡下降路徑）搜索 Cauchy point $x^c$，該點在這條路徑上最小化 $q_k$。然後，它識別出 $x$ 中位於其邊界的分量，並在對其餘分量進行無受限優化 $q_k$ 的過程中保持這些分量不變，以獲得子問題 (48) 的近似解 $\hat{x}$。

# §18.6 Nonlinear Gradient Projection

Cauchy 點 $x^c$ 滿足若投影梯度非零，則 $q_k(x^c) < q_k(x_k)$。由於 Algorithm 16.5 生成了一個子問題解 $\hat{x}$，滿足 $q_k(\hat{x}) \leqslant q_k(x^c)$，我們有以下不等式：

$$f_k = q_k(x_k) > q_k(x^c) \geqslant q_k(\hat{x}) = f_k + \nabla f_k^{\mathrm{T}} p_k + \frac{1}{2} p_k^{\mathrm{T}} B_k p_k.$$

這個不等式暗示 $\nabla f_k^{\mathrm{T}} p_k < 0$，因為假設 $B_k$ 是正定的。

# §18.6 Nonlinear Gradient Projection

我們現在考慮用信賴區域梯度投影法來解決問題 (46)。我們首先構建二次模型 (47)，但由於對 $q_k$ 沒有凸性要求，我們可以將 $B_k$ 定義為 Hessian $\nabla^2 f(x_k)$ 或通過 BFGS 或 SR1 公式獲得的 quasi-Newton 近似。步進量 $p_k$ 是通過求解以下子問題得到的：

$$\min q_k(x) \quad \text{subject to} \quad \ell \leqslant x \leqslant u \text{ and } \|x - x_k\|_\infty \leqslant \Delta_k \tag{49}$$

其中 $\Delta_k > 0$。這個問題可以表述為一個有界限限制的二次規劃問題，如下所示：

$$\min q_k(x) \quad \text{subject to} \quad \max\{\ell, x_k - \Delta_k e\} \leqslant x \leqslant \min\{u, x_k + \Delta_k e\},$$

其中 $e = (1, 1, \cdots, 1)^{\mathrm{T}}$。Algorithm 16.5 可以用來求解這個子問題。步進量 $p_k$ 是否被接受，依據標準的信賴區域策略進行判斷，並根據 $f$ 變化與步進量 $p_k$ 所引起的 $q_k$ 的變化之間的一致性來更新半徑 $\Delta_k$；參見第 4 章。

# §18.6 Nonlinear Gradient Projection

這兩種剛才概述的梯度投影方法每次迭代都需要解決一個不等式限制的二次子問題，因此在形式上是 IQP 方法。然而，它們也可以被視為 EQP 方法，因為它們使用了 Algorithm 16.5 來解決子問題。該算法首先通過找到 Cauchy 點來識別一個工作集合，然後通過將工作集合的限制固定在其界限上來解決一個只具等式限制的子問題。對於大型問題來說，通常使用共軛梯度法來執行子空間最小化（見問題 $(50)_{16}$）。有時需要一個預條件器來使這種方法變得實際；其中最流行的選擇是不完全（和修改的）Cholesky 分解，該方法在 Algorithm 7.3 中有詳細描述。

梯度投影方法原則上可以擴展到更一般的（線性或凸）約束。然而，由於計算投影到一般約束集合的成本高昂，實際實現通常限於有界約束問題 (46)。

# §18.7 Convergence Analysis

數值實驗表明，本章討論的 SQP 和 SLQP 方法通常可以從遠端的起始點收斂到解。因此，人們對於理解是什麼驅使迭代朝向解的方向，以及導致算法失敗的因素感興趣。這些全局收斂性研究對於改進算法的設計和實施非常有價值。

一些早期的研究結果對多重乘子的有界性、子問題 (8) 的良好性質，以及 constraint Jacobian 的正則性等作出了強烈的假設。近年來的研究放寬了許多這些假設，目的是理解迭代過程中成功和失敗的情況。現在我們來陳述一個經典的全局收斂結果，該結果給出了標準 SQP 算法總是識別非線性程序的 KKT 點的條件。

# §18.7 Convergence Analysis

考慮一個 SQP 方法，通過求解二次規劃問題 (8) 來計算搜索方向 $p_k$。我們假設在 (8a) 中，Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 被某個對稱且正定的近似 $B_k$ 替換。新的迭代點定義為 $x_{k+1} + \alpha_k p_k$，其中 $\alpha_k$ 通過一個回溯線搜索來計算，從單位步長開始，並在滿足下列條件時終止：

$$\phi_1(x_k + \alpha_k p_k; \mu) \leqslant \phi_1(x_k; \mu) - \eta\alpha_k(q_\mu(0) - q_\mu(p_k)),$$

其中 $\eta \in (0, 1)$，且 $\phi_1$ 如 (39) 所定義，$q_\mu$ 如 (37) 所定義。為了建立收斂結果，我們假設每個二次規劃問題 (8) 都是可行的並確定了有界解 $p_k$。我們還假設懲罰參數 $\mu$ 對於所有 $k$ 都是固定的且足夠大。

# §18.7 Convergence Analysis

## Theorem

*Suppose that the SQP algorithm just described is applied to the nonlinear program* (7). *Suppose that the sequences* $\{x_k\}$ *and* $\{x_k + p_k\}$ *are contained in a closed, bounded, convex region of* $\mathbb{R}^n$ *in which* $f$ *and* $c_i$ *have continuous first derivatives. Suppose that the matrices* $B_k$ *and multipliers are bounded and that* $\mu$ *satisfies* $\mu \geqslant \|\lambda_k\|_\infty + \rho$ *for all* $k$, *where* $\rho$ *is a positive constant. Then all limit points of the sequence* $\{x_k\}$ *are KKT points of the nonlinear program* (7).

該定理的結論相當令人滿意，但是假設較為嚴格：數列 $\{x_k + p_k\}$ 保持在有界集合內的條件排除了 Hessians $B_k$ 或 constraint Jacobian 變為 ill-conditioned 的情況。定理 19.2 是在更現實的條件下針對非線性內點法建立的全局收斂結果（之一），但類似的結果也可以針對 trust-region SQP 方法建立。

# §18.7 Convergence Analysis

## Theorem

*Suppose that the SQP algorithm just described is applied to the nonlinear program (7). Suppose that the sequences $\{x_k\}$ and $\{x_k + p_k\}$ are contained in a closed, bounded, convex region of $\mathbb{R}^n$ in which $f$ and $c_i$ have continuous first derivatives. Suppose that the matrices $B_k$ and multipliers are bounded and that $\mu$ satisfies $\mu \geqslant \|\lambda_k\|_\infty + \rho$ for all $k$, where $\rho$ is a positive constant. Then all limit points of the sequence $\{x_k\}$ are KKT points of the nonlinear program (7).*

該定理的結論相當令人滿意，但是假設較為嚴格：數列 $\{x_k + p_k\}$ 保持在有界集合內的條件排除了 Hessians $B_k$ 或 constraint Jacobian 變為 ill-conditioned 的情況。定理 19.2 是在更現實的條件下針對非線性內點法建立的全局收斂結果（之一），但類似的結果也可以針對 trust-region SQP 方法建立。

# §18.7 Convergence Analysis

- **Rate of convergence**

我們接下來推導保證 SQP 方法局部收斂的條件，以及確保超線性收斂率的條件。為了簡化討論，我們將討論限制在只具等式限制優化的 Algorithm 18.1 上，並考慮計算 $\nabla^2_{xx}\mathcal{L}_k$ 時使用 exact Hessian 或是 quasi-Newton 兩個不同的版本。這裡提出的結果可以應用於不等式限制問題的算法，一旦 active set 被設定於其最終最優值時（見定理 18.1）。

# §18.7 Convergence Analysis

我們首先列出在本節中將會一直使用的對問題討論的一般假設。

**Assumptions 18.2.**

點 $x_*$ 是問題

$$\min f(x) \quad \text{subject to} \quad c(x) = 0, \tag{1}$$

的局部解，其滿足以下條件：

1. 在 $x_*$ 的某個鄰域內，函數 $f$ 和限制函數 $c$ 具有兩次可微性，且其二次導數是 Lipschitz 連續的。
2. 在 $x_*$ 處滿足 LICQ。這個條件意味著存在某個乘子向量 $\lambda_*$，使得在 $x_*$ 點滿足 KKT 條件（式 $(32)_{12}$）。
3. 在 $(x_*, \lambda_*)$ 處滿足二階充分條件（定理 12.6）。

# §18.7 Convergence Analysis

我們首先考慮一個使用 exact 二階導數的 SQP 方法。

**Theorem**

*Suppose that Assumptions 18.2 hold. Then, if $(x_0, \lambda_0)$ is sufficiently close to $(x_*, \lambda_*)$, the pairs $(x_k, \lambda_k)$ generated by Algorithm 18.1 converge quadratically to $(x_*, \lambda_*)$.*

此定理的證明與定理 11.2 一樣，因為我們知道 Algorithm 18.1 相當於對非線性系統 $F(x, \lambda) = 0$ 應用牛頓法，其中 $F$ 由

$$F(x, \lambda) \equiv \nabla_{(x,\lambda)} \mathcal{L}(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^{\mathrm{T}} \lambda \\ c(x) \end{bmatrix} = 0 \qquad (2)$$

所定義。

# §18.7 Convergence Analysis

我們首先考慮一個使用 exact 二階導數的 SQP 方法。

**Theorem**

*Suppose that Assumptions 18.2 hold. Then, if* $(x_0, \lambda_0)$ *is sufficiently close to* $(x_*, \lambda_*)$*, the pairs* $(x_k, \lambda_k)$ *generated by Algorithm 18.1 converge quadratically to* $(x_*, \lambda_*)$*.*

此定理的證明與定理 11.2 一樣，因為我們知道 Algorithm 18.1 相當於對非線性系統 $F(x, \lambda) = 0$ 應用牛頓法，其中 $F$ 由

$$F(x, \lambda) \equiv \nabla_{(x,\lambda)} \mathcal{L}(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^{\mathrm{T}} \lambda \\ c(x) \end{bmatrix} = 0 \qquad (2)$$

所定義。

# §18.7 Convergence Analysis

我們接下來將目光轉向 Algorithm 18.1 的 quasi-Newton 變體，在這些變體中，Lagrangian Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ 被 quasi-Newton 近似 $B_k$ 取代。我們在 §18.3 討論了使用完整 Hessian 的近似算法，還有保持對投影 Hessian $Z_k^{\mathrm{T}}\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)Z_k$ 近似的 reduced-Hessian 方法。如同先前的討論，我們假設 $Z_k$ 是一個 $n \times (n - m)$ 矩陣，其行 span $A_k$ 的 null space，並額外假設 $Z_k$ 的行是 orthornormal 的；參見 (15.22)。

# §18.7 Convergence Analysis

如果我們將 KKT 系統

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^{\mathrm{T}} \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix} \tag{6}$$

的第一個區塊列乘以 $Z_k$，我們得到

$$Z_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k p_k = -Z_k^{\mathrm{T}} \nabla f_k. \tag{50}$$

當 $x_k$ 和 $\lambda_k$ 不太遠離它們的最優值時，以上方程式與 (6) 的第二個區塊列 $A_k p_k = -c_k$ 一起，足以完全確定 $p_k$ 的值。換句話說，只有 Hessian 的投影 $Z_k^{\mathrm{T}} \nabla^2_{xx}\mathcal{L}_k$ 是顯著的；$\nabla^2_{xx}\mathcal{L}_k$ 的剩餘部分（亦即其在 $A_k^{\mathrm{T}}$ 的值域空間上的投影）不在 $p_k$ 的決定中扮演任何角色。

# §18.7 Convergence Analysis

通過將 (50) 乘以 $Z_k$，並定義下列矩陣 $P_k$，它投影到 $A_k$ 的 null space 上：

$$P_k = \mathrm{I} - A_k^{\mathrm{T}} \left[ A_k A_k^{\mathrm{T}} \right]^{-1} A_k = Z_k Z_k^{\mathrm{T}},$$

我們可以等價地將 (50) 重寫為：

$$P_k \nabla_{xx}^2 \mathcal{L}_k p_k = -P_k \nabla f_k.$$

上述討論結合定理 18.4，表明如果選擇 quasi-Newton 矩陣 $B_k$，使得 $P_k B_k$ 合理逼近 $P_k \nabla_{xx}^2 \mathcal{L}_k$，則 quasi-Newton 方法在局部上是收斂的。如果 $P_k B_k$ 很好地近似 $P_k \nabla_{xx}^2 \mathcal{L}_k$，則方法是超線性收斂的。為了使第二個陳述更加精確，我們介紹了一個結果，可以視為超線性收斂特性（定理 3.6）在等式限制情況下的擴展。在接下來的討論中，$\nabla_{xx}^2 \mathcal{L}_*$ 表示 $\nabla_{xx}^2 \mathcal{L}(x_*, \lambda_*)$。

# §18.7 Convergence Analysis

**Theorem**

*Suppose that Assumptions 18.2 hold and that the iterates $x_k$ generated by Algorithm 18.1 with quasi-Newton approximate Hessians $B_k$ converge to $x_*$. Then $x_k$ converges superlinearly if and only if the Hessian approximation $B_k$ satisfies*

$$\lim_{k\to\infty} \frac{\|P_k(B_k - \nabla^2_{xx}\mathcal{L}_*)(x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \tag{51}$$

我們可以將這個結果應用於本章前面討論過的 quasi-Newton 更新方案，首先是基於 (10) 的完整 BFGS 逼近法。為了保證 BFGS 逼近法始終是 well-defined，我們做出（強）假設，即 Lagrangian 的 Hessian 矩陣在解處是正定的。

# §18.7 Convergence Analysis

**Theorem**

*Suppose that Assumptions 18.2 hold and that the iterates $x_k$ generated by Algorithm 18.1 with quasi-Newton approximate Hessians $B_k$ converge to $x_*$. Then $x_k$ converges superlinearly if and only if the Hessian approximation $B_k$ satisfies*

$$\lim_{k\to\infty} \frac{\|P_k(B_k - \nabla^2_{xx}\mathcal{L}_*)(x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \tag{51}$$

我們可以將這個結果應用於本章前面討論過的 quasi-Newton 更新方案，首先是基於 (10) 的完整 BFGS 逼近法。為了保證 BFGS 逼近法始終是 well-defined，我們做出（強）假設，即 Lagrangian 的 Hessian 矩陣在解處是正定的。

# §18.7 Convergence Analysis

## Theorem

*Suppose that Assumptions 18.2 hold. Assume also that* $\nabla^2_{xx}\mathcal{L}_*$ *and* $B_0$ *are symmetric and positive definite. If* $\|x_0 - x_*\|$ *and* $\|B_0 - \nabla^2_{xx}\mathcal{L}_*\|$ *are sufficiently small, the iterates* $x_k$ *generated by Algorithm 18.1 with BFGS Hessian approximations* $B_k$ *defined by* (10) *and* (12) (*with* $r_k = s_k$) *satisfy the limit* (51). *Therefore, the iterates* $x_k$ *converge superlinearly to* $x_*$.

對於程序 18.2 中給出的阻尼 BFGS 更新策略，我們可以證明其收斂速率是 R-超線性的（而不是通常的 Q-超線性速率；詳見附錄）。

# §18.7 Convergence Analysis

**Theorem**

*Suppose that Assumptions 18.2 hold. Assume also that* $\nabla^2_{xx}\mathcal{L}_*$ *and* $B_0$ *are symmetric and positive definite. If* $\|x_0 - x_*\|$ *and* $\|B_0 - \nabla^2_{xx}\mathcal{L}_*\|$ *are sufficiently small, the iterates* $x_k$ *generated by Algorithm 18.1 with BFGS Hessian approximations* $B_k$ *defined by* (10) *and* (12) (*with* $r_k = s_k$) *satisfy the limit* (51). *Therefore, the iterates* $x_k$ *converge superlinearly to* $x_*$.

對於程序 18.2 中給出的阻尼 BFGS 更新策略，我們可以證明其收斂速率是 R-超線性的（而不是通常的 Q-超線性速率；詳見附錄）。

# §18.7 Convergence Analysis

現在我們考慮更新近似 $M_k$ 到 $Z_k^{\mathrm{T}}\nabla_{xx}^2\mathcal{L}_k Z_k$ 的 reduced-Hessian SQP 方法。從 $P_k$ 的定義中，我們可以看出 $Z_k M_k Z_k^{\mathrm{T}}$ 可以被視為對雙向投影 $P_k\nabla_{xx}^2\mathcal{L}_k P_k$ 的一種近似。由於 reduced-Hessian 方法不近似單邊投影 $P_k\nabla_{xx}^2\mathcal{L}_k$，我們不能期望 (51) 成立。對於這些方法，我們可以通過將 (51) 寫成以下形式來陳述超線性收斂的條件：

$$\lim_{k\to\infty}\left[\frac{P_k(B_k-\nabla_{xx}^2\mathcal{L}_*)P_k(x_{k+1}-x_k)}{\|x_{k+1}-x_k\|} + \frac{P_k(B_k-\nabla_{xx}^2\mathcal{L}_*)(\mathrm{I}-P_k)(x_{k+1}-x_k)}{\|x_{k+1}-x_k\|}\right]=0, \tag{52}$$

其中定義 $B_k = Z_k M_k Z_k^{\mathrm{T}}$。以下結果顯示，僅需第一項趨於零，即可獲得超線性收斂的一個較弱形式，即兩步超線性收斂。

# §18.7 Convergence Analysis

## Theorem

*Suppose that Assumption 18.2 ① holds and that the matrices $B_k$ are bounded. Assume also that the iterates $x_k$ generated by Algorithm 18.1 with approximate Hessians $B_k$ converge to $x_*$, and that*

$$\lim_{k\to\infty} \frac{\|P_k(B_k - \nabla^2_{xx}\mathcal{L}_*)P_k(x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \tag{53}$$

*Then the sequence $\{x_k\}$ converges to $x_*$ two-step superlinearly; that is,*

$$\lim_{k\to\infty} \frac{\|x_{k+2} - x_*\|}{\|x_k - x_*\|} = 0.$$

# §18.7 Convergence Analysis

在使用 BFGS 更新的 reduced-Hessian 方法中，迭代式為

$$x_{k+1} = x_k + Y_k p_Y + Z_k p_Z,$$

其中 $p_Y$ 和 $p_Z$ 由

$$(A_k Y_k) p_Y = -c_k, \tag{14a}$$

$$(Z_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k Z_k) p_Z = -Z_k^{\mathrm{T}} \nabla f_k \tag{16}$$

所給出，其中 (16) 中的 $(Z_k^{\mathrm{T}} \nabla_{xx}^2 \mathcal{L}_k Z_k)$ 被使用校正向量及對稱正定的初始逼近 $M_0$ 的 BFGS 公式所更新的 $M_k$ 所替換。如果我們假設用於定義校正向量的 null space 基底矩陣 $Z_k$ 變化平滑，那麼我們可以應用定理 18.7 來顯示 $x_k$ 以兩步超線性收斂的速率收斂。

# §18.8 Perspectives and Software

SQP 方法在以下情況下效率最高：如果 active constraint 的數量幾乎與變量的數量一樣多，也就是說，自由變量的數量相對較少。與 augmented Lagrangian 方法相比，它們需要更少的函數取值，並且在縮放不良的問題上比下一章描述的非線性內點方法更加穩健。目前尚不清楚在大型問題上，IQP 或 EQP 方法哪一種更有效。目前的研究集中於擴展可以使用 SQP 和 SLQP 方法解決的問題類型。

# §18.8 Perspectives and Software

兩個已建立的 SQP 套件包括 SNOPT [128] 和 FILTERSQP [105]。前者採用 line search 方法，而後者則實現了搭配接受步進量的 filter 之 trust-region 策略。§18.5 中的 SLQP 方法在 KNITRO/ACTIVE [49] 中實現。這三個套件都包含確保子問題始終可行並防止 constraint Jacobian 是 rank-deficient 問題的機制。SNOPT 使用彈性模式（又稱為懲罰模式，參見公式 (9)），在 SQP 子問題不可行或 Lagrange 乘子估計值的範數變得非常大時會啟用此模式。FILTERSQP 包括一個可行性恢復階段，除了促進收斂外，還能快速識別收斂到不可行點的情況。KNITRO/ACTIVE 實施一個使用 Algorithm 18.5 更新策略的懲罰方法。

# §18.8 Perspectives and Software

目前還沒有成熟的 S$\ell_1$QP 方法實現，但原型實現顯示出了潛力。CONOPT [9] 套件實現了通用降低梯度方法以及 SQP 方法。

在實踐中，Lagrangian 的 Hessian 矩陣 $\nabla^2_{xx}\mathcal{L}_k$ 常常使用 quasi-Newton 逼近。BFGS 更新在受限制的問題中通常比在無受限情況下效果差，因為需要維護對應矩陣的正定逼近，而該矩陣通常不具有此性質。然而，SNOPT 和 KNITRO 實現的 BFGS 和有限記憶 BFGS 逼近在實際應用中表現良好。KNITRO 還提供了比 BFGS 選項更有效的 SR1 選項，但如何最佳地實現對受限制優化的全 quasi-Newton 逼近仍需要進一步研究。RSQP 套件 [13] 實現了一個保持對 reduced-Hessian 的 quasi-Newton 逼近的 SQP 方法。

# §18.8 Perspectives and Software

如果忽略 Maratos 效應，使用非平滑 merit 函數或 filter 的優化算法可能會顯著減慢。然而，在實踐中，有選擇性地應用二階校正步驟可以充分解決這些問題。

基於信任區域的梯度投影方法的實現包括 TRON [192] 和 LANCELOT [72]。這兩個代碼使用共軛梯度迭代來執行子空間最小化，並應用不完全的 Cholesky 預條件器。LBFGS-B [322] 和 BLMVM [17] 實現了使用有限記憶 BFGS 更新定義黑塞逼近的梯度投影方法。有限記憶 BFGS 矩陣的特性可用於高效執行投影梯度搜索和子空間最小化。SPG [23] 實現了使用非單調線搜索的梯度投影方法。