最优设计的数学方法

程 极 泰 编著

国防工业出版社

最优设计的数学方法

程极泰　编著

国防工业出版社

内 容 简 介

本书介绍最优设计所涉及到的最优化数学方法，着重介绍近年来常用的单纯形直接解法和它的改型使用的方法；以及序列无约束极小化方法，也就是SUMT解法。

第一章是一些必要的数学基础知识。第二章到第六章是最优设计问题的介绍，以及和最优设计有关的最优化的数学方法。第七章是具体计算和计算程序的介绍，它们的详细计算程序放在书末。

本书可供从事工程设计技术人员，以及科研、教学人员参考。

最优设计的数学方法

程根泰　编著

*

国防工业出版社 出版

新华书店北京发行所发行　　各地新华书店经售

国防工业出版社印刷厂印装

*

850×1168 1/32　印张 8 3/4　221千字

1981年11月第一版　1984年7月第二次印刷　印数：7,701—15,200册

统一书号：15034·2249　　定价：1.10元

前　言

近年来，工程技术中的很多设计问题，越来越多地希望通过适当的数学方法，经由电子计算机的计算，得到最优设计的效果。这种最优设计问题，通常能够表示成一种静态的数学模型，从这种模型可以实现的设计变量中，应用求极值的技巧，就可以得到最优设计解。

在最优化的设计问题中，大量地是数学规划中的非线性数学规划问题。我们曾经和上海船舶运输科学研究所的张燮元等同志一起进行过一些最优设计计算，取得较好的结果。这本《最优设计的数学方法》就是在和有关的几个单位共同实践基础上，结合国内外有关参考文献编写成的。

本书的主要内容是介绍求解非线性规划的直接法和序列无约束极小化方法，并给出这两种方法的详细计算程序。

序列无约束极小化方法要引用变尺度迭代方法，为了使读者能了解第五章介绍的无约束极小化方法的有关数学内容，我们在第一章简要的给出了一些要引用到的线性代数基础知识。为了要清楚说明序列无约束极小化方法在最优设计问题中的作用，我们在第二章中列出一些实例，在第三章中比较了各种数学规划方法，在第四章中讨论了使用这些数学方法的一些准则。第六章是关于第七章计算程序中直接法方面的一些说明。

本书第七章中的两种程序是分别由何焕熹、张克邦二同志试算和应用过的，也是分别由他们作详细的文字介绍的。本书在编写过程中曾得到许多单位的同志大力帮助，在此表示衷心的感谢。

编写这本书，虽然是基于一些实践基础之上，但总的说来，还是在匆忙中完成的，有不妥和错误之处请广大读者批评指正。

目　　录

第一章　预备知识

§1　矩阵概念和矩阵的运算

1. n 维实线性空间 R^n

在一个平面上，选定一点为原点，过原点作两条互相垂直的直线构成坐标轴，在坐标轴上选定单位长以后，就可以把平面上的每一点 p 用一对实数 ξ_1、ξ_2 有序的表为序对（ξ_1，ξ_2），它的第一个元 ξ_1 和第二个元 ξ_2 各是 p 点在第一条（横）轴上和第二条（纵）轴上的投影（见图 1-1）。我们把序对（ξ_1，ξ_2）称为向量，因为它也可以被看成为从原点到 p 点这个径向向量的两个分量。当然，p 点和序对，或者向量，都是不同的对象，但是因为它们之间的关系是很明显的，我们将不加什么区别的，看我们的需要，说它是一点、一个序对或一个向量。

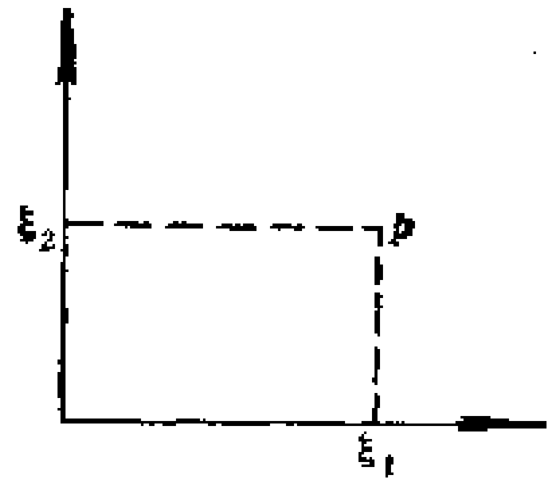


图1-1　平面上 p 点的坐标

这种平面向量的概念可以很自然地推广到一般的实 n 维空间中的向量概念，也就是用一个有序 n 数组表示出一般的向量，定义：

n 向量：顺序排成一列的 n 个实数 ξ_1，ξ_2，…，ξ_n 的整体

$$\mathbf{x}=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} \tag{1-1}$$

称为一个 n 向量 x，实数 $\xi_1, \xi_2, \cdots, \xi_n$ 称为向量 x 的 n 个分量。

在 n 固定时，或者从上下文已经知道 n 的数值时，我们就把 n 向量 x 简称为 x。我们用 R^n 表示所有 n 向量的集合，称这个集合是一个向量空间 R^n，或者简称为空间 R^n。

实数集合表为 R，它的每一个元是一个实数，是一个纯量。但是，向量空间 R^1 的各个元自然是和 R 的各个元对应的，每一个向量

$$(\alpha) \in^{●} R^1$$

相应的有一个纯量

$$\alpha \in R$$

反之亦然。严格地说，空间 R^1 和实数集合 R 是不同的数学对象，但是我们在这里常把它们等价的使用。

两个 n 向量的相等：设 a，b $\in R^n$，且

$$\mathbf{a}=\begin{pmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{pmatrix} \text{ 和 } \mathbf{b}=\begin{pmatrix}\beta_1\\ \beta_2\\ \vdots\\ \beta_n\end{pmatrix} \tag{1-2}$$

只有当它们的对应元都相等时，才叫做相等，也就是当且仅当纯量等式组

$$\alpha_i=\beta_i \qquad i=1,2,\cdots,n \tag{1-3}$$

成立时，才有向量等式

$$\mathbf{a}=\mathbf{b} \tag{1-4}$$

的成立。

两个 n 向量的和：设 a，b $\in R^n$，a，b 的和写成 a＋b，它是一个 n 向量 c，它的各个分量是：

$$\gamma_i=\alpha_i+\beta_i \qquad i=1,2,\cdots,n \tag{1-5}$$

设 a，b，c $\in R^n$，则有加法运算下的交换律和结合律：

（1）a＋b＝b＋a

● $\in$ 表示属于的意思。

（2）$(\mathbf{a}+\mathbf{b})+\mathbf{c}=\mathbf{a}+(\mathbf{b}+\mathbf{c})$

因为，设 $\mathbf{x}=\mathbf{a}+\mathbf{b}$，$\mathbf{y}=\mathbf{b}+\mathbf{a}$，且

$$\mathbf{x}=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} \quad 和 \quad \mathbf{y}=\begin{pmatrix}\eta_1\\ \eta_2\\ \vdots\\ \eta_n\end{pmatrix}$$

则从实数 R 的性质，直接有：

$$\xi_i=\alpha_i+\beta_i=\beta_i+\alpha_i=\eta_i \qquad i=1,2,\cdots,n$$

所以 $\mathbf{x}=\mathbf{y}$，也就是（1）是成立的。再设：

$$\mathbf{x}=(\mathbf{a}+\mathbf{b})+\mathbf{c}$$
$$\mathbf{y}=\mathbf{a}+(\mathbf{b}+\mathbf{c})$$

则有：

$$\xi_i=(\alpha_i+\beta_i)+\gamma_i=\alpha_i+(\beta_i+\gamma_i)=\eta_i$$
$$i=1,2,\cdots,n$$

所以 $\mathbf{x}=\mathbf{y}$，也就是有（2）的性质。

零向量：R^n 中的零向量是 n 个分量都是零的一个 n 向量，这里仍用“$\mathbf{o}$”表示 R^n 中的这个零向量。

设 $\mathbf{a}$ 是 R^n 中的任何一个向量，则它相关于零向量有两个重要性质：

（3）$\mathbf{a}+\mathbf{o}=\mathbf{a}$

（4）在 R^n 中有 $\mathbf{a}$ 的一个逆向量，写成 $(-\mathbf{a})$，它使

$$\mathbf{a}+(-\mathbf{a})=\mathbf{o}$$

因为设 $\mathbf{b}=\mathbf{a}+\mathbf{o}$，则有：

$$\beta_i=\alpha_i+0=\alpha_i \qquad i=1,2,\cdots,n$$

（等式中的 0 是纯量的 0），所以 $\mathbf{b}=\mathbf{a}$，故有（3）。再设

$$\mathbf{b}=\begin{pmatrix}-\alpha_1\\ -\alpha_2\\ \vdots\\ -\alpha_n\end{pmatrix} \quad 且 \quad \mathbf{c}=\mathbf{a}+\mathbf{b}$$

则有：

$$\gamma_i=\alpha_i+\beta_i=\alpha_i+(-\alpha_i)=0 \qquad i=1,2,\cdots,n$$

所以 c=o，并且 b 就是（4）中所要找的逆向量，即

$$-\mathbf{a}=\begin{pmatrix}-\alpha_1\\-\alpha_2\\\vdots\\-\alpha_n\end{pmatrix}=\mathbf{b}$$

这里的符号“－”表示关于向量的一种一元运算。我们还用这个符号表示“减法”的二元运算，也就是，在给定向量 a 和 b 以后，我们用“－”这个符号定义它们的差是：

$$\mathbf{a}-\mathbf{b}=\mathbf{a}+(-\mathbf{b}) \tag{1-6}$$

容易证明，差运算满足纯量之间的减法规律，例如：

$$\mathbf{a}-\mathbf{b}=-(\mathbf{b}-\mathbf{a}) \tag{1-7}$$

纯量和向量的相乘：设 $\lambda\in R$，$\mathbf{a}\in R^n$，则 λ 和 a 的乘积写成 $\lambda\cdot\mathbf{a}$ 或 $\lambda\mathbf{a}$，它是一个 n 向量，它的分量是：

$$\beta_i=\lambda\alpha_i \qquad i=1,2,\cdots,n \tag{1-8}$$

设 λ，$\mu\in R$，a，$\mathbf{b}\in R^n$，则纯量和向量相乘的运算有下列四个性质：

（5）$(\lambda\mu)\mathbf{a}=\lambda(\mu\mathbf{a})$

（6）$(\lambda+\mu)\mathbf{a}=\lambda\mathbf{a}+\mu\mathbf{a}$

（7）$\lambda(\mathbf{a}+\mathbf{b})=\lambda\mathbf{a}+\lambda\mathbf{b}$

（8）$1\cdot\mathbf{a}=\mathbf{a}$

我们有了式（1-8）的定义，就可以把任何一个 n 向量看成是 $1\cdot\mathbf{a}$ 的运算结果，同时 $\lambda\mathbf{a}+\mu\mathbf{b}$ 是表示 $(\lambda\mathbf{a})+(\mu\mathbf{b})$，即先纯量乘再做加法运算，就以（7）的等式来说，则：

设　$\mathbf{x}=\lambda(\mathbf{a}+\mathbf{b}) \qquad \mathbf{y}=\lambda\mathbf{a}+\lambda\mathbf{b}$

因

$$\xi_i=\lambda(\alpha_i+\beta_i)=(\lambda\alpha_i)+(\lambda\beta_i)=\eta_i$$

$$i=1,2,\cdots,n$$

所以（7）是成立的。其余证明类似。

从上面的分析看来，向量空间 R^n 在加法运算，纯量相乘运算，以及零向量和逆向量的存在，给出从（1）到（8）的线性运算性质，所以，我们常常把满足这八个性质的向量空间 R^n 称为线性空间。

一般地说，一个线性空间 S 是指一种集合，它的元不必是由一切 n 个有序实数列成的 n 向量，而可以是在实数（甚至是复数）域上的一个抽象向量空间，只要：

一是对 S 的各个元可以定义一种"+"的加法运算，使它满足（1）（2）的关系，即当 $\mathbf{a},\mathbf{b},\mathbf{c}\in S$ 时，则有：

（1）$\mathbf{a}+\mathbf{b}=\mathbf{b}+\mathbf{a}$

（2）$(\mathbf{a}+\mathbf{b})+\mathbf{c}=\mathbf{a}+(\mathbf{b}+\mathbf{c})$

二是在 S 中有一个"o"元，它使（3）（4）成立，也就是当 $\mathbf{a}\in S$ 时，则有：

（3）$\mathbf{a}+\mathbf{o}=\mathbf{a}$

（4）在 S 中有 $\mathbf{a}$ 元的一个逆元，写成 $(-\mathbf{a})$，它使

$$\mathbf{a}+(-\mathbf{a})=\mathbf{o}$$

三是对 S 的各个元可以给出和实数的一种"·"相乘运算，它使（5）（6）（7）（8）成立，也就是当 $\lambda,\mu\in R,\mathbf{a},\mathbf{b}\in S$ 时，则有：

（5）$(\lambda\mu)\cdot\mathbf{a}=\lambda\cdot(\mu\cdot\mathbf{a})$

（6）$(\lambda+\mu)\cdot\mathbf{a}=\lambda\cdot\mathbf{a}+\mu\cdot\mathbf{a}$

（7）$\lambda\cdot(\mathbf{a}+\mathbf{b})=\lambda\cdot\mathbf{a}+\lambda\cdot\mathbf{b}$

（8）$1\cdot\mathbf{a}=\mathbf{a}$

例如，设 S 是所有定义在（0，1）上的实值函数的集合。若 f，$g\in S$，定义 $h=f+g$ 是一个函数，它的值是：

$$h(\xi)=f(\xi)+g(\xi),\qquad \xi\in(0,1)$$

若 $f\in S$，$\lambda\in R$，定义 $h=\lambda f$ 是一个函数，它的值是：

$$h(\xi)=\lambda f(\xi),\qquad \xi\in(0,1)$$

设"o"是在（0，1）上恒等于零的函数，则 S 将满足（1）

到（8）的性质，所以一切定义在（0，1）上的实函数集合 S 形成一个线性空间，或抽象向量空间。

有了线性空间 S 的基本性质，无论 S 是否是 n 向量空间 R^n 或是更一般的抽象向量空间，总可以根据线性空间的定义导出其它更进一步的性质。例如，若 S 是一个抽象向量空间，且 $\mathbf{a}\in S$，则有：

$$0\cdot\mathbf{a}=\mathbf{o}$$

左边的零是纯量，右边的零是抽象零向量。证明它成立的步骤如下：

$$\begin{aligned}
\mathbf{o}&=\mathbf{a}+(-\mathbf{a}) && (\text{由}(4))\\
&=1\cdot\mathbf{a}+(-\mathbf{a}) && (\text{由}(8))\\
&=(0+1)\cdot\mathbf{a}+(-\mathbf{a}) && (\text{由}(3))\\
&=(0\cdot\mathbf{a}+1\cdot\mathbf{a})+(-\mathbf{a}) && (\text{由}(6))\\
&=0\cdot\mathbf{a}+[1\cdot\mathbf{a}+(-\mathbf{a})] && (\text{由}(2))\\
&=0\cdot\mathbf{a}+[\mathbf{a}+(-\mathbf{a})] && (\text{由}(8))\\
&=0\cdot\mathbf{a}+\mathbf{o} && (\text{由}(4))\\
&=0\cdot\mathbf{a} && (\text{由}(3))
\end{aligned}$$

当然，线性空间最常见的还是 n 向量空间 R^n，这种线性空间又可称为 n 维向量空间，它之所以称为“n 维”，主要的不是因为它是由 n 个有序实数列成的，而是从一般线性空间的“基底”概念来加以说明的。基底的概念是建立在线性空间中的线性独立的概念上的。

2．线性独立向量

设 m 个向量 $\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_m\in R^n$ 和 m 个纯量 $\alpha_1,\alpha_2,\cdots,\alpha_m\in R$ 经过加法运算和纯量相乘的运算，得到新的 n 向量是：

$$\alpha_1\mathbf{x}_1+\alpha_2\mathbf{x}_2+\cdots+\alpha_m\mathbf{x}_m \tag{1-9}$$

则称这个 n 向量是 m 个 n 向量 $\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_m$ 的一个“线性组合”，其中纯量 $\alpha_1,\alpha_2,\cdots,\alpha_m$ 称为线性组合的系数。如果一种组合是由 $\alpha_1=\alpha_2=\cdots=\alpha_m=0$ 构成时，则称线性组合是“平凡的”，否则，

线性组合是一种非平凡的线性组合，简称为"非凡的"。

通常用 $\mathbf{e}^{(i)}$（$i=1,2,\cdots,n$）表示 n 个 n 向量，$\mathbf{e}^{(i)}$ 的第 i 个分量是1，其余的分量都是零。这样，任何一个 n 向量 $\mathbf{x}$ 就可以写成：

$$\mathbf{x}=\xi_1\mathbf{e}^{(1)}+\xi_2\mathbf{e}^{(2)}+\cdots+\xi_n\mathbf{e}^{(n)} \tag{1-10}$$

其中

$$\mathbf{x}=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix},\quad \mathbf{e}^{(1)}=\begin{pmatrix}1\\0\\0\\ \vdots\\0\end{pmatrix},\quad \mathbf{e}^{(2)}=\begin{pmatrix}0\\1\\0\\ \vdots\\0\end{pmatrix},\cdots \mathbf{e}^{(n)}=\begin{pmatrix}0\\0\\0\\ \vdots\\1\end{pmatrix} \tag{1-11}$$

换句话说，任何一个 n 向量总是 $\mathbf{e}^{(1)},\mathbf{e}^{(2)},\cdots,\mathbf{e}^{(n)}$ 这 n 个 n 向量的一个线性组合。

设 X 是 R^n 空间中一部分 n 向量构成的一个集合，称 X 是 R^n 的一个子集，表示成：

$$X\subset R^n$$

如果在 X 中，没有任何的一些元组成的非凡线性组合能够是零向量，则称 X 的元是线性独立的。或者等价地说，若 $\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_m\in X$，且

$$\alpha_1\mathbf{x}_1+\alpha_2\mathbf{x}_2+\cdots+\alpha_m\mathbf{x}_m=\mathbf{o}$$

则 $\alpha_1=\alpha_2=\cdots=\alpha_m=0$，那么，称 X 中的 $\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_m$ 是"线性独立的"。反之，若有不全为零的纯量 $\alpha_1,\alpha_2,\cdots,\alpha_m$ 使 $\alpha_1\mathbf{x}_1+\alpha_2\mathbf{x}_2+\cdots+\alpha_m\mathbf{x}_m=0$，则称 $\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_m$ 是"线性相关的"。

在式（1-11）中给出的 R^n 向量空间中的 n 个向量 $\mathbf{e}^{(1)},\mathbf{e}^{(2)},\cdots,\mathbf{e}^{(n)}$ 是线性独立的向量，因为如果它们的一个线性组合能表出一个零向量 $\mathbf{a}=\mathbf{o}$，即

$$\mathbf{a}=\alpha_1\mathbf{e}^{(1)}+\alpha_2\mathbf{e}^{(2)}+\cdots+\alpha_n\mathbf{e}^{(n)}=\mathbf{o}$$

由于 α_i 就是 $\mathbf{a}$ 的第 i 个分量，它必须是零，所以就有 $\alpha_1=\alpha_2=\cdots=\alpha_n=0$ 的结论。

在 R^n 中，可以由 $m<n$ 个线性独立向量张成一个子空间，也就是定义：

子空间：设 S 是 R^n 的一个非空子集，若

（1） $\mathbf{x}, \mathbf{y} \in S \Longrightarrow \mathbf{x}+\mathbf{y} \in S$

（2） $\mathbf{x} \in S$，$\alpha \in R \Longrightarrow \alpha\mathbf{x} \in S$

则称 S 是 R^n 的一个子空间。这里使用的符号“$\Rightarrow$”表示“若…则…”的语句。

基底：设 $S \subset R^n$ 也就是 S 是 R^n 的一个子空间，且设 $B \subset S$，若

（3） B 的各元是线性独立的；

（4） B 产生了 S，即 B 中各元的任一线性组合都是 S 的元，反之，S 中任何一元都是 B 中各元的一个线性组合。

则称 B 是 S 的一个“基底”。

如果 S 是 R^n 的一个非凡子空间，则 S 的基底所包含的元的个数，称为 S 这个子空间的维。由于 $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \cdots, \mathbf{e}^{(n)}$ 是 R^n 的基底，所以我们说向量空间 R^n 是一个 n 维线性空间。

向量空间的概念是我们讨论矩阵概念的基础。

3. 矩阵

矩阵产生于很多具体的有关问题，最重要的一种直接联系就是用矩阵来表示线性变换。

所谓线性变换，是指一种附合线性运算关系的函数。以从 R^2 到 R^2 的一种函数或映射的简单情况来说，我们用符号

$$f: R^2 \longrightarrow R^2$$

表示一个二自变量，值域是二维的函数或映射，f 把 R^2 中任何一个二向量映射到 R^2 中一个二向量，即

$$\mathbf{y} = f(\mathbf{x}), \quad \mathbf{x} \in R^2, \quad \mathbf{y} \in R^2$$

或者用分量写出是：

$$\eta_1 = f_1(\xi_1, \xi_2)$$
$$\eta_2 = f_2(\xi_1, \xi_2)$$

如果对于任何向量 $\mathbf{x}$，$\mathbf{y} \in R^2$ 和纯量 α，$\beta \in R$，函数或映

射 f 具有下面性质：

$$f(\alpha \mathbf{x}+\beta \mathbf{y})=\alpha f(\mathbf{x})+\beta f(\mathbf{y}) \tag{1-12}$$

则称函数 f 是一个线性变换。

事实上，线性变换 f 是完全由它在基底向量 $\mathbf{e}^{(1)}$ 和 $\mathbf{e}^{(2)}$ 上的数值唯一决定的。也就是，设

$$f(\mathbf{e}^{(1)})=\mathbf{a}^{(1)}=\begin{pmatrix}\alpha_{11}\\ \alpha_{21}\end{pmatrix},\quad f(\mathbf{e}^{(2)})=\mathbf{a}^{(2)}=\begin{pmatrix}\alpha_{12}\\ \alpha_{22}\end{pmatrix} \tag{1-13}$$

则对任何的 $\mathbf{x}=\xi_1\mathbf{e}^{(1)}+\xi_2\mathbf{e}^{(2)}$，从（1-12）可以有：

$$\begin{aligned}\mathbf{y}=f(\mathbf{x})&=f(\xi_1\mathbf{e}^{(1)}+\xi_2\mathbf{e}^{(2)})=\xi_1 f(\mathbf{e}^{(1)})+\xi_2 f(\mathbf{e}^{(2)})\\&=\xi_1\mathbf{a}^{(1)}+\xi_2\mathbf{a}^{(2)}\end{aligned}$$

也就是，用分量表出为：

$$\eta_1=\alpha_{11}\xi_1+\alpha_{12}\xi_2,\quad \eta_2=\alpha_{21}\xi_1+\alpha_{22}\xi_2$$

它完全由 2 × 2 的阵列（即二行二列的阵列）

$$\begin{pmatrix}\alpha_{11} & \alpha_{12}\\ \alpha_{21} & \alpha_{22}\end{pmatrix} \tag{1-14}$$

决定。（1-14）这个阵列就称为表示线性变换 f 的矩阵。

一般地说，如果给出一种映射

$$f: R^n \longrightarrow R^m$$

对于任何向量 $\mathbf{x}$，$\mathbf{y}\in R^n$ 和纯量 α，$\beta\in R$ 也有线性变换性质（1-12），则表示线性变换 f 的对应矩阵将是由 m 行 n 列的一个 $m\times n$ 矩阵

$$\mathbf{A}=\begin{pmatrix}\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n}\\ \cdots & \cdots & & \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn}\end{pmatrix} \tag{1-15}$$

表示，阵列中的 $m\times n$ 个实数（或在其它需要的情况下是复数）α_{ij}（$i=1,2,\cdots,m$，$j=1,2,\cdots,n$）称为矩阵 A 的元。

所有 $m\times n$ 矩阵的集合用 $R^{m\times n}$ 表示。

矩阵（1-15）右边的表示方法，也有用方括弧的，写成：

$$\mathbf{A}=\begin{bmatrix}\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n}\\ \cdots & \cdots & & \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn}\end{bmatrix}$$

更多地是用罗马小体字母代替希腊字母写出矩阵的各元，也就是：

$$\mathbf{A}=\begin{pmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \cdots & \cdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn}\end{pmatrix} \tag{1-16}$$

并且常常简称为 $m\times n$ 矩阵，写成 $\mathbf{A}=(a_{ij})$。

我们常常把 $n\times 1$ 矩阵

$$\mathbf{C}=\begin{pmatrix}a_{11}\\ a_{21}\\ \vdots\\ a_{n1}\end{pmatrix} \tag{1-17}$$

看成是和一个 n 向量的同一表达形式，从而在 $R^{n\times 1}$ 和 R^n 之间就不加什么区别。

类似地，我们称 $1\times n$ 矩阵

$$\mathbf{R}=(a_{11}, a_{12}, \cdots, a_{1n}) \tag{1-18}$$

是一个行向量，而且在式（1-18）中二元之间加上逗号主要是防止书写上混乱，并且一个行向量并不用来和 n 维空间的坐标点对应，我们仍然是以列向量（1-17）和点做等价对应的描述。

在 $m\times n$ 矩阵（1-16）中，若 $m=n$，则称 $\mathbf{A}$ 是一个 n 阶方阵。如果一个方阵的非零元仅在对角元 a_{ii}（$i=1,2,\cdots,n$）上出现，则称这个方阵是一个对角阵，写成：

$$\begin{pmatrix}\delta_1 & 0 & \cdots & 0\\ 0 & \delta_2 & \cdots & 0\\ \cdots & \cdots & & \\ 0 & 0 & \cdots & \delta_n\end{pmatrix}=\mathrm{diag}(\delta_1, \delta_2, \cdots, \delta_n) \tag{1-19}$$

如果一个 n 阶方阵 $\mathbf{A}$ 的各元有

$$a_{ij}=a_{ji} \qquad i, j=1, 2, \cdots, n \tag{1-20}$$

的性质时，则称为一个对称矩阵。

我们常常看具体的需要将一个矩阵分块为子矩阵表示出来，例如一个 4×4 的矩阵

$$\mathbf{A}=\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

可用 2×2 的子矩阵写成分块矩阵的形式，即

$$\mathbf{A}=\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}$$

其中

$$\mathbf{A}_{11}=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad \mathbf{A}_{12}=\begin{pmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{pmatrix}$$

$$\mathbf{A}_{21}=\begin{pmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{pmatrix}, \quad \mathbf{A}_{22}=\begin{pmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{pmatrix}$$

设 $\mathbf{A}\in R^{m\times n}$，$\mathbf{A}$ 的第 j 列是 m 向量的列向量，即

$$\mathbf{a}^{(j)}=\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \qquad j=1, 2, \cdots, n$$

则 $\mathbf{A}$ 可以用各列向量分块写成：

$$\mathbf{A}=(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \cdots, \mathbf{a}^{(n)})$$

类似地，若 $\mathbf{A}$ 的第 i 行是行向量：

$$\mathbf{b}^{(i)}=(a_{i1}, a_{i2}, \cdots, a_{in}) \qquad i=1, 2, \cdots, m$$

则 $\mathbf{A}$ 可以用各行向量分块写成：

$$\mathbf{A}=\begin{pmatrix} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \\ \vdots \\ \mathbf{b}^{(m)} \end{pmatrix}$$

设 **A** 是一个 $m \times n$ 矩阵，将其行列互换形成的矩阵

$$\mathbf{B}=\begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \cdots & \cdots & & \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{pmatrix}$$

称为 **A** 的转置矩阵，简称为 **A** 的转置，写成：

$$\mathbf{B}=\mathbf{A}^T$$ ●

此时

$$b_{ij}=a_{ji} \qquad i=1, 2, \cdots, n, j=1, 2, \cdots, m$$

一个 n 向量 **x** 可视为一个 $n \times 1$ 列矩阵或列向量，即

$$\mathbf{x}=\begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix}$$

它的转置 $\mathbf{x}^T$ 是一个行向量，或 $1 \times n$ 行矩阵，即

$$\mathbf{x}^T=(\xi_1, \xi_2, \cdots, \xi_n)$$

反之，常常为了节省篇幅起见，我们把列向量用行向量表示为：

$$\mathbf{x}=(\xi_1, \xi_2, \cdots, \xi_n)^T$$

4．矩阵的运算

我们从线性变换引出了矩阵概念，我们将根据线性变换的性质引出矩阵运算的定义。

设 f 和 g 是从 R^2 到 R^2 的线性变换，并且相应的有一个新函数（新映射或新变换），即

$$h: R^2 \to R^2$$

此函数称为 f 和 g 的和，定义为：

● 有时，也可写成 $\mathbf{B}=\mathbf{A}'$，为了避免和求导相混，我们将采用 T 表示转置。

$$h(\mathbf{x})=f(\mathbf{x})+g(\mathbf{x}) \qquad \mathbf{x}\in R^2 \tag{1-21}$$

由于

$$\begin{aligned}h(\alpha\mathbf{x}+\beta\mathbf{y})&=f(\alpha\mathbf{x}+\beta\mathbf{y})+g(\alpha\mathbf{x}+\beta\mathbf{y})\\&=\alpha f(\mathbf{x})+\beta f(\mathbf{y})+\alpha g(\mathbf{x})+\beta g(\mathbf{y})\\&=\alpha[f(\mathbf{x})+g(\mathbf{x})]+\beta[f(\mathbf{y})+g(\mathbf{y})]\\&=\alpha h(\mathbf{x})+\beta h(\mathbf{y})\end{aligned}$$

对任何 $\mathbf{x},\mathbf{y}\in R^2$，$\alpha,\beta\in R$是成立的，所以 h 是一个线性变换。

我们知道任何一个线性变换都可以用一个矩阵唯一的表示，设 f，g 和 h 的矩阵表示各为：

$$\mathbf{A}=(\mathbf{a}^{(1)},\ \mathbf{a}^{(2)}),\qquad \mathbf{B}(\mathbf{b}^{(1)},\ \mathbf{b}^{(2)}),\qquad \mathbf{C}=(\mathbf{c}^{(1)},\ \mathbf{c}^{(2)})$$

而每个矩阵的列向量是：

$$\mathbf{a}^{(i)}=f(\mathbf{e}^{(i)}),\qquad \mathbf{b}^{(i)}=g(\mathbf{e}^{(i)}),\qquad \mathbf{c}^{(i)}=h(\mathbf{e}^{(i)})$$
$$i=1,\ 2$$

所以，从式（1-21）得：

$$\mathbf{c}^{(i)}=h(\mathbf{e}^{(i)})=f(\mathbf{e}^{(i)})+g(\mathbf{e}^{(i)})=\mathbf{a}^{(i)}+\mathbf{b}^{(i)}$$
$$i=1,\ 2$$

也就是用 A，B，C 的各元写出的关系是：

$$c_{ij}=a_{ij}+b_{ij} \qquad i,j=1,\ 2 \tag{1-22}$$

另一方面，设 $f: R^2\to R^2$ 是一个线性变换，且 $\lambda\in R$，则定义函数 $g=\lambda f$ 是：

$$g(\mathbf{x})=\lambda f(\mathbf{x}) \tag{1-23}$$

所以容易看出 g 也是一个线性变换，并且，如果 A 和 B 是表示 f 和 g 的矩阵，则有：

$$b_{ij}=\lambda a_{ij} \qquad i,\ j=1,\ 2 \tag{1-24}$$

从线性变换的上述概念和分析，我们可以类似于 R^n 中的 n 向量的"相等"，"相加"，"零向量"，"纯量相乘"的定义，给出 $R^{m\times n}$ 中的 $m\times n$ 矩阵的"相等"，"相加"，"零矩阵"，"纯量和矩阵相乘"的定义如下：

两个 $m\times n$ 矩阵的相等：设 A，B$\in R^{m\times n}$，且

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad B=\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & & \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} \tag{1-25}$$

只有当它们的对应元都相等时，才叫做相等，也就是当且仅当纯量等式组

$$a_{ij}=b_{ij} \qquad i=1,\ 2,\cdots,\ m,\quad j=1,\ 2,\cdots,\ n \tag{1-26}$$

成立时，才有矩阵等式

$$A=B \tag{1-27}$$

的成立。

两个 $m\times n$ 矩阵相加的和：设 A，B$\in R^{m\times n}$，且是式（1-25）的形式，则 A，B 的和写成 A+B，它是一个 $m\times n$ 矩阵C，C 的各个元是：

$$c_{ij}=a_{ij}+b_{ij} \qquad i=1,2,\cdots,m,\quad j=1,\ 2,\cdots,n \tag{1-28}$$

这个加法定义是和线性变换式（1-22）的要求一致的。

设 A，B，C 是 $m\times n$ 矩阵，则有加法运算下的交换律和结合律：

（1）A+B=B+A

（2）(A+B)+C=A+(B+C)

因为，从它们的各个元的关系，直接可以有：

$$a_{ij}+b_{ij}=b_{ij}+a_{ij} \qquad i=1,\ 2,\cdots,\ m,\quad j=1,2,\cdots,\ n$$

$$(a_{ij}+b_{ij})+c_{ij}=a_{ij}+(b_{ij}+c_{ij}) \qquad i=1,\ 2,\cdots,m,$$

$$j=1,2,\cdots,\ n$$

所以有（1）（2）的结果。

零矩阵：$R^{m\times n}$ 中的零矩阵是 mn 个元都是零的一个 $m\times n$ 矩阵，用“0”这个符号表示 $R^{m\times n}$ 中的这个零矩阵。

设 A 是 $R^{m\times n}$ 中的任一个 $m\times n$ 矩阵，则零矩阵有性质：

（3） $\mathbf{A}+\mathbf{O}=\mathbf{A}$

（4） 有一个矩阵 $-\mathbf{A}$ 使 $\mathbf{A}+(-\mathbf{A})=\mathbf{O}$

（3）是显然的，至于（4），只要取 $-\mathbf{A}$ 的各元是 $-a_{ij}$ 就有 $\mathbf{A}+(-\mathbf{A})=\mathbf{O}$ 的结果。

这里的一元运算符号“$-$”也可以派生出“减法”的二元运算，也就是在任给两个 $m\times n$ 矩阵 A, B 以后，我们用“$-$”这个符号给出 A 与 B 之差的定义是：

$$\mathbf{A}-\mathbf{B}=\mathbf{A}+(-\mathbf{B}) \tag{1-29}$$

并且，也容易证明，矩阵的差运算也满足纯量之间减法的一些规律，例如：

$$\mathbf{A}-\mathbf{B}=-(\mathbf{B}-\mathbf{A}) \tag{1-30}$$

这里的一元符号“$-$”，是否就等价于 -1 这个纯量相乘的运算结果，还有赖于下列定义：

纯量和矩阵的相乘：设 A 是一个 $m\times n$ 矩阵，且 $\lambda\in R$，则 λ 和 A 的乘积写成 $\lambda\cdot\mathbf{A}$ 或 $\lambda\mathbf{A}$，它是 $m\times n$ 矩阵，$\lambda\mathbf{A}$ 的元是：

$$\lambda a_{ij} \qquad i=1,\ 2,\cdots,m,\quad j=1,\ 2,\cdots,n \tag{1-31}$$

这个定义是针对线性变换的性质式(1-23)和式（1-24）给出的。用式（1-31）的定义，容易证得：

设 A 和 B 是 $m\times n$ 矩阵，λ，$\mu\in R$，则有：

（5） $(\lambda\mu)\mathbf{A}=\lambda(\mu\mathbf{A})$

（6） $(\lambda+\mu)\mathbf{A}=\lambda\mathbf{A}+\mu\mathbf{A}$

（7） $\lambda(\mathbf{A}+\mathbf{B})=\lambda\mathbf{A}+\lambda\mathbf{B}$

（8） $1\cdot\mathbf{A}=\mathbf{A}$

从这些性质就可以看出一元运算“$-$”的意思等价于：

$$-\mathbf{A}=(-1)\mathbf{A}=(-1)\cdot 1\cdot\mathbf{A} \tag{1-32}$$

从以上所说明的一切 $m\times n$ 矩阵组成的 $R^{m\times n}$ 空间具有（1）～（8）的性质，这是和向量空间类似，$R^{m\times n}$ 的任何元在加法运算和纯量相乘的运算下也构成了一种线性空间。

线性空间的形成是我们进一步给出它们更多概念和特性的基础，有了这些发展，我们才有可能利用已有的数学方法引导到很多更为有用的方面。例如，逼近的意义和其误差估计的方法等。

矩阵除了在加法和纯量相乘下构成线性空间之外，根据线性变换性质我们还可以给出矩阵和矩阵之间相乘的定义。

设 f，g 都是 $R^2 \to R^2$ 的线性变换，则 f 和 g 的合成变换是指函数，即

$$h(\mathbf{x}) = g[f(\mathbf{x})] \quad \mathbf{x} \in R^2 \tag{1-33}$$

由于

$$\begin{aligned} h(\alpha\mathbf{x}+\beta\mathbf{y}) &= g[f(\alpha\mathbf{x}+\beta\mathbf{y})] = g[\alpha f(\mathbf{x})+\beta f(\mathbf{y})] \\ &= \alpha g[f(\mathbf{x})]+\beta g[f(\mathbf{y})] \\ &= \alpha h(\mathbf{x})+\beta h(\mathbf{y}) \end{aligned}$$

对任何 $\mathbf{x}$，$\mathbf{y} \in R^2$，α，$\beta \in R$ 成立，所以 h 是一个线性变换。

设 f，g 和 h 的表示矩阵各为 $\mathbf{A}$，$\mathbf{B}$ 和 $\mathbf{C}$，则有：

$$\mathbf{c}^{(i)} = h(\mathbf{e}^{(i)}) = g[f(\mathbf{e}^{(i)})] = g(\mathbf{a}^{(i)})$$
$$i = 1, \ 2$$

仿照（1-14），按各列分量写出是：

$$\begin{aligned} c_{11} &= b_{11}a_{11}+b_{12}a_{21} \\ c_{21} &= b_{21}a_{11}+b_{22}a_{21} \\ c_{12} &= b_{11}a_{12}+b_{12}a_{22} \\ c_{22} &= b_{21}a_{12}+b_{22}a_{22} \end{aligned}$$

合写成：

$$c_{ij} = b_{i1}a_{1j}+b_{i2}a_{2j} \qquad i, \ j = 1, \ 2 \tag{1-34}$$

所以，我们应该如下定义：

矩阵的相乘：设 $\mathbf{A}$ 是一个 $l \times m$ 矩阵，$\mathbf{B}$ 是一个 $m \times n$ 矩阵，则 $\mathbf{A}$ 和 $\mathbf{B}$ 的乘积是一个 $l \times n$ 矩阵 $\mathbf{C}$，它的元是：

$$c_{ij} = \sum_{k=1}^{m} a_{ik}b_{kj} \qquad i = 1, 2, \cdots, l, \quad j = 1, 2, \cdots, n \tag{1-35}$$

并且写成：

$$C = AB$$

由此可见，AB 定义的成立，只有在 A 的列个数等于 B 的行个数时，才有意义。所以，一般的情况，矩阵相乘的运算不满足交换律，但是在乘法有意义的情况下，它有如下的一些规律：

$$(AB)C = A(BC)$$
$$A(B+C) = AB+AC$$
$$(A+B)C = AC+BC$$
$$\lambda(AB) = (\lambda A)B = A(\lambda B)$$

特别是，如果 A 是 $l \times m$ 矩阵，B 是 $m \times n$ 矩阵，则 $(AB)^T$ 将是一个 $l \times n$ 矩阵的转置矩阵，所以是一个 $n \times l$ 矩阵；另外 B^T 是 $n \times m$ 矩阵，A^T 是 $m \times l$ 矩阵，B^TA^T 可以成立，并且也是一个 $n \times l$ 矩阵，我们可以证明有下列重要等式：

$$(AB)^T = B^TA^T \tag{1-36}$$

事实上，设 $C = (AB)^T$，由于 AB 的 (j, i) 元是：

$$\sum_{k=1}^{m} a_{jk}b_{ki}$$

所以它应该是 C 的 (i, j) 元，即

$$c_{ij} = \sum_{k=1}^{m} a_{jk}b_{ki}$$

设用 a'_{ij}，b'_{ij} 表示 A^T 和 B^T 各 (i, j) 元，用 d_{ij} 表示 B^TA^T 的 (i, j) 元，则：

$$d_{ij} = \sum_{k=1}^{m} b'_{ik}a'_{kj} = \sum_{k=1}^{m} b_{ki}a_{jk} = \sum_{k=1}^{m} a_{jk}b_{ki} = c_{ij}$$

所以知道 $B^TA^T = C$，也就是式（1-36）成立。

容易看出，对于任何一个矩阵 A，矩阵 A^TA 和 AA^T 总有定义，并且总是对称的，因为：

$$(A^TA)^T = A^T(A^T)^T = A^TA$$

5．线性方程组

一类线性变换逆问题的出现，就是在很多理论上和实用上提出的对于未知数 $\xi_1, \xi_2, \cdots, \xi_n$ 或 n 向量 $\mathbf{x}=(\xi_1, \xi_2, \cdots, \xi_n)^T$ 求解线性方程组

$$\begin{aligned} a_{11}\xi_1+a_{12}\xi_2+\cdots+a_{1n}\xi_n&=\beta_1 \\ a_{21}\xi_1+a_{22}\xi_2+\cdots+a_{2n}\xi_n&=\beta_2 \\ \cdots\cdots \\ a_{m1}\xi_1+a_{m2}\xi_2+\cdots+a_{mn}\xi_n&=\beta_m \end{aligned} \tag{1-37}$$

的问题。其中：常数项 $\beta_1, \beta_2, \cdots, \beta_m$ 是一个给定的 m 向量 $\mathbf{b}=(\beta_1, \beta_2, \cdots, \beta_m)^T$。式（1-37）常用矩阵形式简写为：

$$\mathbf{A}\mathbf{x}=\mathbf{b} \tag{1-38}$$

设 $\mathbf{A}$ 是一个 $m\times n$ 矩阵，由 $\mathbf{A}$ 的各列的线性组合所张成的 R^m 中的子空间，称为 $\mathbf{A}$ 的"列空间"，写成 $R(\mathbf{A})$，而 $\mathbf{A}$ 的行空间可以由 $\mathbf{A}$ 的转置 $\mathbf{A}^T$ 的列空间来定义，也就是 $\mathbf{A}$ 的"行空间"是：

$$R(\mathbf{A}^T)\subset R^n$$

我们称 $\mathbf{A}$ 的列空间的维为 $\mathbf{A}$ 的"秩"，写成 $\mathrm{rank}(\mathbf{A})$，也就是：

$$\mathrm{rank}(\mathbf{A})=\dim[R(\mathbf{A})] \tag{1-39}$$

一切使式（1-38）对应的齐次方程组

$$\mathbf{A}\mathbf{x}=0 \tag{1-40}$$

成立的 $\mathbf{x}$ 的集合，称为 $\mathbf{A}$ 的"零空间"，写成 $N(\mathbf{A})$，它的维数 $\dim[N(\mathbf{A})]$ 称为 $\mathbf{A}$ 的"零度"，写成 $\mathrm{Null}(\mathbf{A})$。

在这些概念的基础上，我们就有线性方程组求解的下列定理：

定理 1　当且仅当增广矩阵 $(\mathbf{A},\mathbf{b})$ 的秩等于 $\mathbf{A}$ 的秩时，即

$$\mathrm{rank}[(\mathbf{A},\mathbf{b})]=\mathrm{rank}(\mathbf{A}) \tag{1-41}$$

时，式（1-37）有一解 $\mathbf{x}$ 存在。

证：设有一个 x 使 $\mathbf{A}\mathbf{x}=\mathbf{b}$，则 b 是 A 的各列的一个线性组合，所以 b 仍在 $R(\mathbf{A})$ 中，也就是 $R[(\mathbf{A},\mathbf{b})]=R(\mathbf{A})$，所以：

$$\operatorname{rank}[(\mathbf{A},\mathbf{b})]=\operatorname{rank}(\mathbf{A})$$

反之，由于

$$R(\mathbf{A})\subset R[(\mathbf{A},\mathbf{b})]$$

且又因（1-41）成立，$R(\mathbf{A})$ 和 $R[(\mathbf{A},\mathbf{b})]$ 具有相同维数，所以 $R(\mathbf{A})=R[(\mathbf{A},\mathbf{b})]$，也就是 $\mathbf{b}\in R(\mathbf{A})$，从而 b 是 A 的各列的一个线性组合，所以有 $\mathbf{x}\in R^n$ 使 $\mathbf{b}=\mathbf{A}\mathbf{x}$。证毕。

定理 2　若 $\operatorname{rank}(\mathbf{A})=m$，则式（1-37）总有一解。

证：由于 A 和 $(\mathbf{A},\mathbf{b})$ 的各列都是 m 向量，而且：

$$\operatorname{rank}(\mathbf{A})\leqslant\operatorname{rank}[(\mathbf{A},\mathbf{b})]\leqslant m$$

因 $\operatorname{rank}(\mathbf{A})=m$，所以有：

$$\operatorname{rank}(\mathbf{A})=\operatorname{rank}[(\mathbf{A},\mathbf{b})]=m$$

从式（1-41）可知式（1-37）总有一解。证毕。

定理 3　设 x 是式（1-37）的一解，则式（1-37）所有解的集合是：

$$\mathbf{x}+N(\mathbf{A})\doteq\{\mathbf{x}+\mathbf{z}:\ \mathbf{z}\in N(\mathbf{A})\}\tag{1-42}$$

证：若 y 满足式（1-37），则从

$$\mathbf{A}\mathbf{x}=\mathbf{b},\qquad \mathbf{A}\mathbf{y}=\mathbf{b}$$

相减得 $\mathbf{A}(\mathbf{y}-\mathbf{x})=0$，所以 $(\mathbf{y}-\mathbf{x})\in N(\mathbf{A})$，并且：

$$\mathbf{y}=\mathbf{x}+(\mathbf{y}-\mathbf{x})\in\mathbf{x}+N(\mathbf{A})$$

反之，若 $\mathbf{y}\in\mathbf{x}+N(\mathbf{A})$，则有某 z 合于 $\mathbf{A}\mathbf{z}=0$ 使 $\mathbf{y}=\mathbf{x}+\mathbf{z}$，所以 $\mathbf{A}\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{A}\mathbf{z}=\mathbf{A}\mathbf{x}=\mathbf{b}$，y 是式（1-37）的解。证毕。

按照上述定理，可以看出，欲使式（1-37）有唯一解，必须 $\operatorname{Null}(\mathbf{A})=0$，所以应该 $m\geqslant n$。这时如果对应的逆线性变换关系是能用

$$\mathbf{x}=\mathbf{B}\mathbf{b}\tag{1-43}$$

表出时，这自然是指：

$$\mathbf{B}\mathbf{A}\mathbf{x}=\mathbf{B}\mathbf{b},\qquad \mathbf{B}\mathbf{A}=\mathbf{I}$$

其中：$\mathbf{I}$是一个n阶单位矩阵，即

$$\mathbf{I}=\mathrm{diag}\{1,1,\cdots,1\}=\begin{pmatrix}1&0&\cdots&0\\0&1&\cdots&0\\\cdots&\cdots&\cdots&\cdots\\0&0&\cdots&1\end{pmatrix}\tag{1-44}$$

$\mathbf{B}$是一个$n\times m$矩阵。但是如果

$$\mathbf{A}\mathbf{x}=\mathbf{b}$$

对于任何$\mathbf{b}\in R^m$都能按式（1-43）解出，必然是$\mathbf{A}$的n列能张成R^m，那么它只在$m=n$时才能成立。这时，我们称n阶方阵$\mathbf{B}$是$\mathbf{A}$的逆阵，写成$\mathbf{A}^{-1}$。从式（1-38）和式（1-43）可以看出逆阵满足的关系是：

$$\mathbf{A}\mathbf{A}^{-1}=\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}\tag{1-45}$$

这样的n阶方阵$\mathbf{A}$称为是“非奇异的”，简称非奇。

容易看出，若$\mathbf{A}$是非奇方阵，则$\mathbf{A}^{-1}$也是非奇方阵，并且：

$$(\mathbf{A}^{-1})^{-1}=\mathbf{A}\tag{1-46}$$

因为

$$(\mathbf{A}^{-1})(\mathbf{A}^{-1})^{-1}=\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}$$
$$(\mathbf{A}^{-1})^{-1}(\mathbf{A}^{-1})=\mathbf{A}\mathbf{A}^{-1}=\mathbf{I}$$

如果$\mathbf{A}$是非奇方阵，则$\mathbf{A}^T$也是非奇方阵，并且：

$$(\mathbf{A}^T)^{-1}=(\mathbf{A}^{-1})^T\tag{1-47}$$

因为

$$(\mathbf{A}^T)(\mathbf{A}^T)^{-1}=\mathbf{A}^T(\mathbf{A}^{-1})^T=(\mathbf{A}^{-1}\mathbf{A})^T=\mathbf{I}^T=\mathbf{I}$$
$$(\mathbf{A}^T)^{-1}\mathbf{A}^T=(\mathbf{A}^{-1})^T\mathbf{A}^T=(\mathbf{A}\mathbf{A}^{-1})^T=\mathbf{I}^T=\mathbf{I}$$

最后指出：设$\mathbf{A}$，$\mathbf{B}$是n阶方阵，则当且仅当$\mathbf{A}$和$\mathbf{B}$非奇时，$\mathbf{AB}$才是非奇，并且：

$$(\mathbf{AB})^{-1}=\mathbf{B}^{-1}\mathbf{A}^{-1}\tag{1-48}$$

事实上，若$\mathbf{A}$和$\mathbf{B}$非奇，则$\mathbf{B}^{-1}\mathbf{A}^{-1}$是$\mathbf{AB}$的逆，因为：

$$\mathbf{B}^{-1}\mathbf{A}^{-1}(\mathbf{AB})=\mathbf{B}^{-1}(\mathbf{A}^{-1}\mathbf{A})\mathbf{B}=\mathbf{B}^{-1}\mathbf{I}\mathbf{B}=\mathbf{B}^{-1}\mathbf{B}=\mathbf{I}$$
$$(\mathbf{AB})\mathbf{B}^{-1}\mathbf{A}^{-1}=\mathbf{A}(\mathbf{B}\mathbf{B}^{-1})\mathbf{A}^{-1}=\mathbf{A}\mathbf{I}\mathbf{A}^{-1}=\mathbf{A}\mathbf{A}^{-1}=\mathbf{I}$$

反之，设 AB 是非奇，$(AB)^{-1}$ 存在，并且从上所述，可知 $(AB)^{-1}=B^{-1}A^{-1}$，则 $B(AB)^{-1}$ 是 A 的逆，所以 A 非奇；$(AB)^{-1}A$ 是 B 的逆，所以 B 非奇。因为：

$$\{B(AB)^{-1}\}A=BB^{-1}A^{-1}A=I$$

$$A\{B(AB)^{-1}\}=ABB^{-1}A^{-1}=I$$

同时

$$\{(AB)^{-1}A\}B=B^{-1}A^{-1}AB=I$$

$$B\{(AB)^{-1}A\}=BB^{-1}A^{-1}A=I$$

在我们结束矩阵概念和矩阵的运算这一节的时候，我们再举两个以后要用得到的推算例子：

（1）设 $A\in R^{m\times n}$，则当且仅有某非零 $x\in R^m$ 和某非零 $y\in R^n$ 使 $A=xy^T$ 时，$\mathrm{rank}(A)=1$。也就是 A 为秩 1 阵的充分必要条件是：

$$x\in R^m,\ y\in R^n \text{ 使 } A=xy^T$$

证：若有 $x=(\xi_1,\ \xi_2,\cdots,\ \xi_m)^T$，　$y=(\eta_1,\ \eta_2,\cdots,\ \eta_n)^T$ 使：

$$A=xy^T=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_m\end{pmatrix}(\eta_1,\ \eta_2,\cdots,\ \eta_n)=\begin{pmatrix}\xi_1\eta_1 & \xi_1\eta_2 & \cdots & \xi_1\eta_n\\ \xi_2\eta_1 & \xi_2\eta_2 & \cdots & \xi_2\eta_n\\ \cdots\cdots & \cdots\cdots & \cdots & \cdots\\ \xi_m\eta_1 & \xi_m\eta_2 & \cdots & \xi_m\eta_n\end{pmatrix} \tag{1-49}$$

也就是 $A=(a^{(1)},a^{(2)},\cdots,a^{(n)})$ 中 $a^{(2)},a^{(3)},\cdots,a^{(n)}$ 各列向量是列向量 $a^{(1)}$ 的线性组合，所以 A 的列空间只有一个独立列向量，故

$$\mathrm{rank}(A)=\dim[R(A)]=1$$

反之，若 $\mathrm{rank}(A)=1$，则 $a^{(2)},\ a^{(3)},\cdots,\ a^{(n)}$ 是 $a^{(1)}$ 的线性组合，故有式（1-49）的从右到左的结果，所以有非零向量 $x\in R^m$，$y\in R^n$ 使 $A=xy^T$。

（2）设 $B\in R^{n\times n}$，$S\in R^{k\times k}$ 且 $U,\ V\in R^{n\times k}$，又设 B，S，$V^TB^{-1}U-S^{-1}$ 非奇，则有：

$$(B-USV^T)^{-1}=B^{-1}-B^{-1}UTV^TB^{-1} \tag{1-50}$$

其中 $S^{-1}+T^{-1}=V^TB^{-1}U$

证：因为

$$\begin{aligned}&(B-USV^T)(B^{-1}-B^{-1}UTV^TB^{-1})\\&=I-USV^TB^{-1}-UTV^TB^{-1}+USV^TB^{-1}UTV^TB^{-1}\\&=I-U[(S+T)-SV^TB^{-1}UT]V^TB^{-1}\\&=I-U[(S+T)-S(S^{-1}+T^{-1})T]V^TB^{-1}\\&=I-U[(S+T)-(S+T)]V^TB^{-1}\\&=I-0=I\end{aligned}$$

显然，同样可证得：

$$(B^{-1}-B^{-1}UTV^TB^{-1})(B-USV^T)=I$$

§2 矩阵的范数和映射的导数

1. 矩阵的特征值

我们在前面已经指出 R^n 是表示实数分量为 ξ_1, ξ_2, …, ξ_n的列向量 x 的实 n 维线性空间。对应的，我们用 C^n 表示复数分量的列向量空间。在 $x\in R^n$ 时，x^T 表示 x 的转置，而当 $x\in C^n$，我们采用的对应关系是共轭转置 x^H，也就是，如果 x 的分量 ξ_1, ξ_2, …, ξ_n 是复数，则有：

$$x^H=(\bar{\xi}_1,\ \bar{\xi}_2,\cdots,\ \bar{\xi}_n) \tag{1-51}$$

这里的 $\bar{\xi}_i$ 是 ξ_i 的共轭。总之，x^T，x^H 都是行向量。

一个实 $m\times n$ 矩阵 $A=(a_{ij})$ 定义出一个从 R^n 到 R^m 的线性映射，写成 $A\in R^{m\times n}$，或者更清楚的写成：

$$A\in L(R^n, R^m) \tag{1-52}$$

从而，它既说明了矩阵类型，也反映了它表示的线性变换的一种“线性算子”的作用。这种矩阵表示和线性映射的不必加以区别的前题是 R^n 中采用的基底向量是：

$e^{(1)}=(1,\ 0,\ 0,\cdots,\ 0)^T$, $e^{(2)}=(0,1,0,\cdots,0)^T$,

……, $e^{(n)}=(0,\ 0,\ 0,\cdots,\ 1)^T$

在$n=m$时，我们把$L(R^n,R^n)$简写为$L(R^n)$。如果$\mathbf{A}$有逆阵$\mathbf{A}^{-1}$，则称线性算子$\mathbf{A}\in L(R^n)$或$\mathbf{A}\in L(C^n)$是可逆的或非奇的。并且用$\mathbf{A}^T$或$\mathbf{A}^H$表示$\mathbf{A}$的转置矩阵和共轭转置矩阵。$\mathbf{A}^T$的各元是$\mathbf{A}$中各元行列互相转置，$\mathbf{A}^H$的各元是$\mathbf{A}$中各元行列互相转置并取其共轭。

对于任一$\mathbf{A}\in L(C^n)$，如果一个复数λ能使方程

$$\mathbf{A}\mathbf{x}=\lambda\mathbf{x} \tag{1-53}$$

有一个非零解$\mathbf{x}$，则称λ是$\mathbf{A}$的一个"特征值"，非零解$\mathbf{x}$称为$\mathbf{A}$对应于λ的一个"特征向量"。

如果$\mathbf{A}$是一个实对称矩阵，也就是：

$$\mathbf{A}=\mathbf{A}^T \tag{1-54}$$

这时$\mathbf{A}$的一切特征值都是实的，并且对于任何$\mathbf{x}\in R^n$有：

$$\lambda_1\mathbf{x}^T\mathbf{x}\leqslant\mathbf{x}^T\mathbf{A}\mathbf{x}\leqslant\lambda_n\mathbf{x}^T\mathbf{x} \tag{1-55}$$

其中：$\lambda_1\leqslant\lambda_2\leqslant\cdots\leqslant\lambda_n$是$\mathbf{A}$的特征值。

若$\mathbf{A}\in L(R^n)$对任何一个$\mathbf{x}\in R^n$有

$$\mathbf{x}^T\mathbf{A}\mathbf{x}\geqslant 0 \tag{1-56}$$

时，称$\mathbf{A}$这个n阶方阵是一个"半正定的矩阵"。如果在$\mathbf{x}\neq\mathbf{o}$时，

$$\mathbf{x}^T\mathbf{A}\mathbf{x}>0 \tag{1-57}$$

恒成立，则称$\mathbf{A}$是一个"正定矩阵"。任何一个正定矩阵$\mathbf{A}$总是有逆阵$\mathbf{A}^{-1}$。

2．向量的范数和矩阵的范数

一个从R^n(或C^n)到R^1的映射，表示为$\|\cdot\|$，如果满足：

$$\left.\begin{array}{l}
(1)\ \|\mathbf{x}\|\geqslant 0,\quad \forall^{*}\ \mathbf{x}\in R^n\ (\text{或}\ C^n),\\
\qquad\qquad \text{且仅当}\ \mathbf{x}=\mathbf{o}\ \text{时},\|\mathbf{x}\|=0。\\
(2)\ \|\alpha\mathbf{x}\|=|\alpha|\ \|\mathbf{x}\|,\quad \forall\ \mathbf{x}\in R^n(\text{或}\ C^n),\\
\qquad\qquad \alpha\in R^1(\text{或}C^1)。\\
(3)\ \|\mathbf{x}+\mathbf{y}\|\leqslant\|\mathbf{x}\|+\|\mathbf{y}\|,\quad \forall\ \mathbf{x},\ \mathbf{y}\in R^n(\text{或}C^n),
\end{array}\right\} \tag{1-58}$$

则称非负实数$\|\cdot\|$是一个"范数"。

---

* 符号$\forall$表示"所有"的意思。

给出范数的目的可以使我们比较 R^n(或C^n) 中各元之间接近的情况。因为两个元接近是要用距离来说明的，距离是一种长度概念，有了范数的定义，我们就可以描述 x, y∈R^n二元之间的距离是：

$$\rho(x, y) \triangleq^{●} \|x-y\| \tag{1-59}$$

规定范数应满足的三个条件是使我们能就向量空间 R^n (或C^n)，甚至是一般的线性空间中元与元之间大小和距离进行计算和分析。

在 R^n 或 C^n 中的范数的选用例子是 l_p 范数：

$$\|x\|_p = \left(\sum_{i=1}^{n} |\xi_i|^p\right)^{1/p} \qquad 1 \leqslant p < \infty \tag{1-60}$$

以及它的极限情况，l_∞ 范数：

$$\|x\|_\infty = \max_{1\leqslant i\leqslant n} |\xi_i| \tag{1-61}$$

这种范数概念显然是空间向量长度概念的一般化，特别是 l_2 范数，即

$$\|x\|_2 = \left(\sum_{i=1}^{n} |\xi_i|^2\right)^{1/2} \tag{1-62}$$

就是普通欧几里得几何中距离概念的直接推广，所以我们把用式(1-62) 范数规范的线性空间，称为是一般的欧氏空间。

向量在空间中有长度或大小的概念以外，还有内积(数乘积)的概念，向量长度是一元转化为非负实数的关系，内积则是二元转化为某对应实数或复数的关系，对于实向量空间来说内积是一个实数。

一般地说，在 R^n 上的一个“内积”，表示为$(\cdot,\cdot)$，它满足：

$$\left.\begin{array}{ll} (1) & (x,\ x) \geqslant 0,\ \forall x\in R^n;\ 且仅当\ x=o\ 时,\\ & (x,\ x)=0。\\ (2) & (x,\ y)=(y,\ x),\ \forall x,\ y\in R^n。\end{array}\right\} \tag{1-63}$$

● “≜”表示定义ρ(x, y)为x, y二元之间的距离的符号，下同。

$$
\left.\begin{array}{l}
(3)\ (x+y,\ z)=(x,\ z)+(y,\ z),\ (\alpha x,\ y) \\
\quad =\alpha(x,\ y),\ \forall x,\ y,\ z\in R^n,\ \ \alpha\in R^1。
\end{array}\right\}
$$

在 R^n 上有了内积概念以后，自然就可以直接得到范数概念，此时：

$$\|x\|=(x,\ x)^{1/2} \tag{1-64}$$

特别是 l_2 范数（也就是欧氏范数），它实际上是从内积定义

$$(x,\ y)=x^T y \tag{1-65}$$

导出的。

对于任何一种内积，都有柯西-舒瓦尔兹（Cauchy-Schwarz）不等式，即

$$|(x,\ y)|\leqslant\|x\|\,\|y\| \tag{1-66}$$

式（1-66）再取 l_2 范数时就是：

$$\left|\sum_{i=1}^{n}\xi_i\eta_i\right|\leqslant\left(\sum_{i=1}^{n}|\xi_i|^2\right)^{1/2}\left(\sum_{i=1}^{n}|\eta_i|^2\right)^{1/2} \tag{1-67}$$

关于 R^n 上内积的概念自然可以推广到复向量空间 C^n，这时只要在式（1-63）的（2）改用

$$(x,\ y)=(\overline{y,\ x}) \tag{1-68}$$

就可以了。

其次，我们来考虑线性算子的范数。设在 R^n 和 R^m 上各给定任二范数 $\|\cdot\|$ 和 $\|\cdot\|'$，并给定任一个矩阵 $A\in L(R^n,\ R^m)$，则 A 对于 $\|\cdot\|$ 和 $\|\cdot\|'$ 的算子范数定义为：

$$\|A\|=\sup_{\|x\|=1}\|Ax\|' \tag{1-69}$$

这样一个矩阵范数满足下列三个性质：

$$
\left.\begin{array}{l}
(1)\ \|A\|\geqslant 0,\ \forall A\in L(R^n,R^m),\text{且仅当 } A=0 \text{ 时，} \\
\qquad \|A\|=0。\\
(2)\ \|\alpha A\|=|\alpha|\,\|A\|,\ \ \forall A\in L(R^n,\ R^m),\ \ \alpha\in R^1。\\
(3)\ \|A+B\|\leqslant\|A\|+\|B\|,\ \ \forall A,\ B\in L(R^n,\ R^m)。
\end{array}\right\} \tag{1-70}
$$

一般的说，我们把满足式（1-58）或式（1-70）的线性空间

都称为一个"赋范线性空间"，所以向量空间 R^n 或 $m\times n$ 矩阵集合 $L(R^n, R^m)$ 都是可以看成一种赋范线性空间。

特别重要的情况是 $R^n=R^m$ 且 $\|\cdot\|$ 和 $\|\cdot\|'$ 相同时，就更有

$$\|\mathbf{AB}\|\leqslant\|\mathbf{A}\|\,\|\mathbf{B}\| \tag{1-71}$$

的关系。

设 $\mathbf{A}\in L(R^n, R^m)$，并且 R^n 和 R^m 上的范数都以 l_i 范数 $(i=1, 2, \infty)$ 给出，则：

$$\|\mathbf{A}\|_1=\max_{1\leqslant j\leqslant n}\sum_{i=1}^{m}|a_{ij}| \tag{1-72}$$

$$\|\mathbf{A}\|_\infty=\max_{1\leqslant i\leqslant m}\sum_{j=1}^{n}|a_{ij}| \tag{1-73}$$

$$\|\mathbf{A}\|_2=\sqrt{\lambda} \tag{1-74}$$

其中：λ 是 $\mathbf{A}^T\mathbf{A}$ 的最大特征值。

证：关于 l_1 范数，对于任一 $\mathbf{x}\in R^n$ 有：

$$\|\mathbf{Ax}\|_1=\sum_{i=1}^{m}\left|\sum_{j=1}^{n}a_{ij}x_j\right|\leqslant\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|\,|x_j|$$

$$\leqslant\sum_{j=1}^{n}|x_j|\sum_{i=1}^{m}|a_{ij}|\leqslant\left(\max_{1\leqslant j\leqslant n}\sum_{i=1}^{m}|a_{ij}|\right)\|\mathbf{x}\|_1$$

所以，只要指出对于某 $\mathbf{x}\in R^n$ 能有等式成立就可以了。设 k 是使式（1-72）极大的附标，则：

$$\|\mathbf{Ae}^{(k)}\|_1=\sum_{i=1}^{m}|a_{ik}|=\max_{1\leqslant j\leqslant n}\sum_{i=1}^{m}|a_{ij}|$$

也就是式（1-69）中的上确界对第 k 个坐标向量成立。

类似地，在 l_∞ 情况，上确界在 R^n 上的 $\mathbf{x}$ 各分量是：

$$\xi_i=\begin{cases}a_{ki}/|a_{ki}|, & a_{ki}\neq 0\\ 1\quad , & a_{ki}=0\end{cases}\qquad i=1, 2, \cdots, n$$

其中：k 是使式（1-73）极大的一个附标。

最后，关于 l_2 范数，因为

$$\|\mathbf{A}\mathbf{x}\|_2=(\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x})^{1/2}$$

所以从式（1-55）关于实对称阵的性质，立即可以得到式（1-74）的结果。证毕。

3．一般映射的导数

我们已经知道，一个单变量的实值函数 f 在一个 x 点可微的意思是指有一个实数 $a=f'(\mathrm{x})$ 使

$$\lim_{t\to 0}(1/t)[f(\mathrm{x}+t)-f(\mathrm{x})-at]=0 \qquad (1\text{-}75)$$

把这种导数概念推论到一般 n 个变量 m 个实值函数 f 的情况，设 f 的定义域是 R^n 中某一个 D 域，则一般的函数 f 用映射写出就是：

$$f: D\subset R^n\to R^m \qquad (1\text{-}76)$$

如果对于 D 的一个内点 x，有一个线性算子 $A\in L(R^n, R^m)$ 对于任一个 $\mathbf{h}\in R^n$ 能使

$$\lim_{t\to 0}(1/t)\|f(\mathrm{x}+t\mathbf{h})-f(\mathrm{x})-t\mathbf{A}\mathbf{h}\|=0 \qquad (1\text{-}77)$$

成立，则称 f 是在 D 的内点 x “可微”，更清楚地说是“伽杜(Gateaux)可微”，或简称“G 可微”。

事实上，可以证明式（1-77）的极限存在是与 R^m 上所采用的具体范数无关，也就是，如果 f 在 x 就某范数是 G 可微的时候，f 在 x 就可以对任何一种范数 G 可微。

如果 f：$D\subset R^n\to R^m$ 是在 D 的内点 x G 可微，使式（1-77）成立的线性算子 $\mathbf{A}\in L(R^n, R^m)$ 表为 $f'(\mathrm{x})$，称为 f 在 x 的“G 导数”。

设 f 的分量函数是 f_1，$f_2, \cdots, f_m$，它们各是 n 元的实值函数，从而合成定义出 f 的 m 值函数。研究这种情况下 $f'(\mathrm{x})$ 的具体表达形式，设 $\mathbf{A}=(a_{ij})$，$\mathbf{h}$ 选为第 j 个坐标向量 $\mathbf{e}^{(j)}$，则式(1-77)给出：

$$\lim_{t\to 0}(1/t)|f_i(\mathrm{x}+t\mathbf{e}^{(j)})-f_i(\mathrm{x})-ta_{ij}|=0$$

上式表示 f_i 在 x 的偏导数存在，并且

$$\partial_j f_i(\mathrm{x}) \equiv \partial f_i(\mathrm{x})/\partial x_j = a_{ij} \qquad i=1,\ 2,\cdots,m,$$
$$j=1,\ 2,\cdots,\ n \qquad (1-78)$$

所以在 f 有 G 导数 $f'(\mathrm{x})$ 时，它就可以用雅可比矩阵表为：

$$f'(\mathrm{x})=\begin{pmatrix} \dfrac{\partial f_1(\mathrm{x})}{\partial x_1} & \dfrac{\partial f_1(\mathrm{x})}{\partial x_2} & \cdots & \dfrac{\partial f_1(\mathrm{x})}{\partial x_n} \\ \dfrac{\partial f_2(\mathrm{x})}{\partial x_1} & \dfrac{\partial f_2(\mathrm{x})}{\partial x_2} & \cdots & \dfrac{\partial f_2(\mathrm{x})}{\partial x_n} \\ \cdots & \cdots & \cdots & \\ \dfrac{\partial f_m(\mathrm{x})}{\partial x_1} & \dfrac{\partial f_m(\mathrm{x})}{\partial x_2} & \cdots & \dfrac{\partial f_m(\mathrm{x})}{\partial x_n} \end{pmatrix} \qquad (1-79)$$

特别是，如果给出一个 n 元函数，也就是给出一个泛函形式的映射，即

$$F: D \subset R^n \to R^1$$

则 $F'(\mathrm{x})=f(\mathrm{x})$ 可以用一个行向量函数表出为：

$$F'(\mathrm{x})=\left(\frac{\partial F(\mathrm{x})}{\partial x_1},\ \frac{\partial F(\mathrm{x})}{\partial x_2},\cdots,\frac{\partial F(\mathrm{x})}{\partial x_n}\right) \qquad (1-80)$$

它就是 $F(\mathrm{x})$ 在 x 的梯度向量函数的转置，也就是：

$$F'(\mathrm{x})=[\mathrm{grad}F(\mathrm{x})]^T$$

其中

$$\mathrm{grad}F(\mathrm{x})=\nabla_F(\mathrm{x})=\begin{pmatrix} \dfrac{\partial F(\mathrm{x})}{\partial x_1} \\ \dfrac{\partial F(\mathrm{x})}{\partial x_2} \\ \vdots \\ \dfrac{\partial F(\mathrm{x})}{\partial x_n} \end{pmatrix} \qquad (1-81)$$

另一方面，如果把梯度函数（1-81）做为一个 $R^n \to R^n$ 的映射，并写成：

$$f(\mathrm{x}) \triangleq \nabla_F(\mathrm{x})=[F'(\mathrm{x})]^T \qquad (1-82)$$

则从式（1-79）的 G 导数定义，如果 $f(\mathrm{x})$ 是 G 可微时，就有：

$$f'(x)=\begin{pmatrix} \dfrac{\partial^2 F(x)}{\partial x_1^2} & \dfrac{\partial^2 F(x)}{\partial x_2\partial x_1} & \cdots & \dfrac{\partial^2 F(x)}{\partial x_n\partial x_1} \\ \dfrac{\partial^2 F(x)}{\partial x_1\partial x_2} & \dfrac{\partial^2 F(x)}{\partial x_2^2} & \cdots & \dfrac{\partial^2 F(x)}{\partial x_n\partial x_2} \\ \cdots & \cdots & \cdots & \\ \dfrac{\partial^2 F(x)}{\partial x_1\partial x_n} & \dfrac{\partial^2 F(x)}{\partial x_2\partial x_n} & \cdots & \dfrac{\partial^2 F(x)}{\partial x_n{}^2} \end{pmatrix} \tag{1-83}$$

若用 $H_F(x)$ 表示式（1-83）右边的矩阵，它称为是 $F(x)$ 的一个赫森（Hessian）矩阵，所以 $F(x)$ 的赫森矩阵是 $F(x)$ 的梯度函数的 G 导数。

第二章　最优化问题

§1　概　述

在很多工程设计问题中，常常要选用一组“设计参数”，在一定条件之下，使那些不希望出现的“效应”尽可能地小；或者使总的设计方案的费用最少；有时，给出一种设计“优良”的标准，使这个标准数值尽可能地大。这样，我们就将采用最优化的数学方法来协助完成这种设计工作，给出一种最优设计。

所以，最优化的数学方法是完成最优化问题的一种手段。特别是由于现在的电子计算机逐渐地在各个部门普遍使用，最优化的数学方法写成计算机的语言，就可以使最优设计的问题带有了最优手段的性质，形成自动化最优设计的双重最优意义。

一个最优化问题的提出，包括三个方面的正确选定：设计变量，目标函数，约束条件。

首先，谈谈设计变量，它是在设计问题中出现的各个可以选择取值的变动参数。譬如，在船舶设计中，船长、船宽、型深、吃水、方型系数（这些习惯上各用L、B、H、T与C_B表出）等主尺度的设计变量，和船舶载重量D_{WT}、船速率V_S这些设计要素的设计变量，如果这些设计变量都要考虑，我们将用数学上独立自变量的符号表出：

$$\left.\begin{array}{ll} x_1=L & x_5=C_B \\ x_2=B & x_6=D_{WT} \\ x_3=H & x_7=V_S \\ x_4=T & \end{array}\right\} \qquad (2\text{-}1)$$

合并写成变向量的表示为：

$$\mathbf{x}=\begin{pmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\end{pmatrix} \text{或} \ \mathbf{x}=(x_1, x_2, x_3, x_4, x_5, x_6, x_7)^T$$

一般地说，如果选用的设计变量有 n 个时，就可以用一个 n 向量 $\mathbf{x}$ 表出是：

$$\mathbf{x}=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix} \text{或} \ \mathbf{x}=(x_1, x_2, \cdots, x_n)^T \qquad (2\text{-}2)$$

这样，一方面设计变量用一个列向量，或一个行向量的转置表示出来，从而把 n 个设计变量等价地表示为一个 $n\times 1$ 矩阵；另一方面，由于设计变量 x_1, x_2, …, x_n 通常定义在一个 n 维欧氏空间中，或者说各变量有一定的定义域，即

$$a_i\leqslant x_i\leqslant b_i \qquad i=1, 2, \cdots, n \qquad (2\text{-}3)$$

其中：a_i, b_i ($i=1, 2, \cdots, n$) 通常又都是实数，并且测量两组设计变量或两个 n 向量设计变量

$$\mathbf{x}^{(1)}=(x_1^{(1)}, x_2^{(1)}, \cdots, x_n^{(1)})^T, \qquad \mathbf{x}^{(2)}=(x_1^{(2)}, x_2^{(2)}, \cdots, x_n^{(2)})^T$$

之间的距离度量是用一般二维，三维常用的欧几里得几何距离概念的推广形式：

$$\|\mathbf{x}^{(1)}-\mathbf{x}^{(2)}\|\triangleq \mathrm{P}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})=\left[\sum_{i=1}^{n}(x_i^{(1)}-x_i^{(2)})^2\right]^{1/2} \qquad (2\text{-}4)$$

所以，我们说 n 向量设计变量 $\mathbf{x}$ 是取值于一个 n 维欧氏空间的一个定义域 D 内，或简称变量 $\mathbf{x}$ 的定义域是 D，写成：

$$\mathbf{x}\in D\subset R^n \qquad (2\text{-}5)$$

更为一般性地说，设计变量 $\mathbf{x}$ 的定义域 D 还是 n 向量的一个子空间，也就是说．对于 D 中任何两个 n 向量 $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$，以及任

何两个纯量α，β，它们所构成的线性组合$\alpha\mathbf{x}^{(1)}+\beta\mathbf{x}^{(2)}$仍在定义域$D$中，详细写出来就是：

若
$$\mathbf{x}^{(1)}=\begin{pmatrix}x_1^{(1)}\\x_2^{(1)}\\\vdots\\x_n^{(1)}\end{pmatrix}\in D,\quad \mathbf{x}^{(2)}=\begin{pmatrix}x_1^{(2)}\\x_2^{(2)}\\\vdots\\x_n^{(2)}\end{pmatrix}\in D$$

则
$$\alpha\mathbf{x}^{(1)}+\beta\mathbf{x}^{(2)}=\begin{pmatrix}\alpha x_1^{(1)}\\\alpha x_2^{(1)}\\\vdots\\\alpha x_n^{(1)}\end{pmatrix}+\begin{pmatrix}\beta x_1^{(2)}\\\beta x_2^{(2)}\\\vdots\\\beta x_n^{(2)}\end{pmatrix}=\begin{pmatrix}\alpha x_1^{(1)}+\beta x_1^{(2)}\\\alpha x_2^{(1)}+\beta x_2^{(2)}\\\vdots\\\alpha x_n^{(1)}+\beta x_n^{(2)}\end{pmatrix}\in D \tag{2-6}$$

这样的一个n向量子空间D可以看成是一般n维线性向量空间R^n的一个子集，写成：

$$\mathbf{x}\in D\subset R^n \tag{2-7}$$

我们之所以把R^n称为是“n维”的线性向量空间的原因，一方面是因为如果任何两个$\mathbf{x}^{(1)}$，$\mathbf{x}^{(2)}\in R^n$的n向量和任何两个纯量的线性组合将和式（2-6）所表示的那样仍在R^n中，即

$$\text{若}\mathbf{x}^{(1)},\ \mathbf{x}^{(2)}\in R^n,\ \text{则}\alpha\mathbf{x}^{(1)}+\beta\mathbf{x}^{(2)}\in R^n。$$

另一方面，由所有实数为向量分量构成的n向量的总体集合R^n，它包含有n个线性独立的基底向量，即

$$\mathbf{e}^{(1)}=\begin{pmatrix}1\\0\\0\\\vdots\\0\end{pmatrix},\quad \mathbf{e}^{(2)}=\begin{pmatrix}0\\1\\0\\\vdots\\0\end{pmatrix},\quad\cdots,\quad \mathbf{e}^{(n)}=\begin{pmatrix}0\\0\\0\\\vdots\\1\end{pmatrix} \tag{2-8}$$

使R^n中任何一个向量$\mathbf{x}=(x_1,\ x_2,\ \cdots,\ x_n)^T$能表达成基底向量组（2-8）的一个线性组合，即

$$\mathbf{x}=x_1\mathbf{e}^{(1)}+x_2\mathbf{e}^{(2)}+\cdots+x_n\mathbf{e}^{(n)} \tag{2-9}$$

说式（2-8）的 n 个 n 向量是线性独立的意思，就是不可能找到不全为零的 n 个实数 k_1，k_2，…，k_n能使

$$k_1\mathbf{e}^{(1)}+k_2\mathbf{e}^{(2)}+\cdots+k_n\mathbf{e}^{(n)}=\mathbf{o}$$

反过来说，在 R^n 中总可以找出 n 个线性独立的基底向量，而且不必只是式（2-8）这样最简单的基底向量组。

另外，一个线性向量空间 R^n，对其中任何两个 n 向量能给出的距离的量不必一定就是式（2-4）所定义的欧氏范数。关于线性向量空间 R^n 的这些性质，已经在第一章中作了简单地介绍。

其次，让我们来说明目标函数的确定。

目标函数是待最优化的目标，它是 n 个设计变量 x_1，x_2，…，x_n 的一个实函数．或者说是一个 n 向量 $\mathbf{x}$ 的函数，可写成：

$$F(\mathbf{x})=F(x_1, x_2, \cdots, x_n),\quad \mathbf{x}\in D\subset R^n \tag{2-10}$$

有时为了更清楚地阐明，称这种纯量函数是从 R^n 到 R^1 的一个映射，写成：

$$F: D\subset R^n\to R^1 \tag{2-11}$$

这是指我们能够用一个目标实数值通过设计变量选取可以得到最优值的情况。更为一般的是目标函数有若干个，或者说目标函数是从 R^n 到 R^m 的映射，即

$$F: D\subset R^n\to R^m \tag{2-12}$$

目前的最优化数学方法多数是针对单一目标函数来求解的，多目标函数的问题还正在发展和探讨之中。

就以单一目标函数来说，选取不同的标准，目标函数的最优化数值就可以不一样。

譬如，在 R^2 空间中，如果以距离函数做为目标函数，那么就可以按不同的距离定义而有不同的函数形式。设在 n 维线性向量空间 R^n 中给出范数定义是以下几种之一：

$$\left.\begin{aligned}
&\|\mathbf{x}\|_1=\sum_{i=1}^{n}|x_i| \\
&\|\mathbf{x}\|_2=\left[\sum_{i=1}^{n}(x_i)^2\right]^{1/2} \\
&\|\mathbf{x}\|_\infty=\max_{1\leqslant i\leqslant n}|x_i| \\
&\|\mathbf{x}\|_{(M)}=(\mathbf{x},\ \mathbf{M}\mathbf{x})^{1/2}=(\mathbf{x}^T,\ \mathbf{M}\mathbf{x})^{1/2}=\left(\sum_{i,j=1}^{n}m_{ij}x_ix_j\right)^{1/2}
\end{aligned}\right\}\quad(2\text{-}13)$$

其中：M 是一个 $n\times n$ 对称正定矩阵，即

$$\mathbf{M}=(m_{ij})=\begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n}\\ m_{21} & m_{22} & \cdots & m_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ m_{n1} & m_{n2} & \cdots & m_{nn}\end{pmatrix}$$

它使 x 只有当 $\mathbf{x}=0$ 时才有 $\|\mathbf{x}\|_{(M)}=0$。并且式（2-13）的第四种范数定义是通过内积定义

$$(\mathbf{x},\ \mathbf{y})=x_1y_1+x_2y_2+\cdots+x_ny_n \qquad (2\text{-}14)$$

给出的关于二次型

$$\mathbf{x}^T\mathbf{M}\mathbf{x}=\sum_{i,j=1}^{n}m_{ij}x_ix_j \qquad (2\text{-}15)$$

的一种正定二次型的平方根表示出来的。

按式（2-13）四种范数的定义，我们就可以给 R^n 空间中任何两点之间的距离定义是下列四种之一：

$$\left.\begin{aligned}
&\|\mathbf{x}-\mathbf{y}\|_1=P_1(\mathbf{x},\ \mathbf{y})=\sum_{i=1}^{n}|x_i-y_i| \\
&\|\mathbf{x}-\mathbf{y}\|_2=P_2(\mathbf{x},\ \mathbf{y})=\left[\sum_{i=1}^{n}(x_i-y_i)^2\right]^{1/2} \\
&\|\mathbf{x}-\mathbf{y}\|_\infty=P_\infty(\mathbf{x},\ \mathbf{y})=\max_{1\leqslant i\leqslant n}|x_i-y_i| \\
&\|\mathbf{x}-\mathbf{y}\|_M=P_{(M)}(\mathbf{x},\ \mathbf{y})=(\mathbf{x}-\mathbf{y},\ \mathbf{M}(\mathbf{x}-\mathbf{y}))^{1/2}
\end{aligned}\right\}\quad(2\text{-}16)$$

用上式表示出来的 R^2 空间距离目标函数的等值曲线为：

$$F_1(\mathbf{x})=\sum_{i=1}^{2}|x_i|=|x_1|+|x_2|=C_1$$

$$F_2(\mathbf{x})=\left(\sum_{i=1}^{2}x_i^2\right)^{1/2}=(x_1^2+x_2^2)^{1/2}=C_2$$

$$F_3(\mathbf{x})=\max_{1\leqslant i\leqslant 2}|x_i|=\max(|x_1|,\ |x_2|)=C_3$$

$$F_4(\mathbf{x})=(\mathbf{x},\ \mathbf{Mx})^{1/2}=\left(\sum_{i,j=1}^{2}m_{ij}x_ix_j\right)^{1/2}$$

$$=(m_{11}x_1^2+2m_{12}x_1x_2+m_{22}x_2^2)^{1/2}=C_4$$

详见图 2-1 中的（a）（b）（c）（d）。

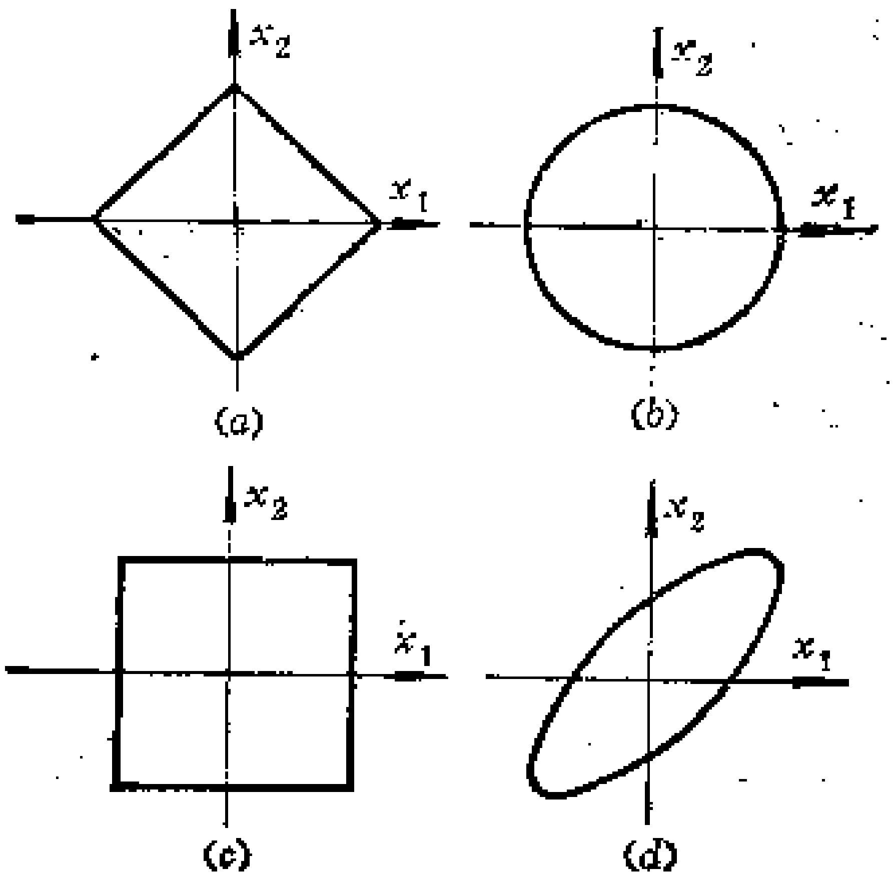


图2-1　四种不同的距离定义给出的四种等值曲线

再例如对于未知函数

$$y=f(x)\qquad x,\ y\in R$$

给出采样点 $y_i=f(x_i)$　（$i=1,\ 2,\ \cdots,\ n$）数据以后，用线性函数 $ax+b$ 进行回归，我们可以采用加权误差平方和

$$S(a,\ b)=\sum_{i=1}^{n} w_i(y_i-ax_i-b)^2$$

作为待极小化的目标函数，这样的最优化方法就是最小二乘方法。或者、以极大误差

$$A(a,\ b)=\max_{1\leqslant i\leqslant n}|y_i-ax_i-b|$$

作为目标函数，选取 a，b 使 $A(a,\ b)$ 达极小值，就是一种契比雪夫逼近。

最后，谈谈关于最优化问题的约束条件，这是最为复杂同时又是要善于处理的问题。一般地说，就 R^n 空间中 D 域内取值的设计变量 n 向量 x 而言，除了已确定好目标函数（2-11）以外，还可能要求所求目标函数的最优解适合某些等式约束条件：

$$h_i(\mathbf{x})=0 \qquad i=1,\ 2,\cdots,\ m \tag{2-17}$$

以及不等式约束条件：

$$g_i(\mathbf{x})\geqslant 0 \qquad i=m+1,\cdots,\ p \tag{2-18}$$

严格来说，等式约束的给出，反映了设计变量的某些不独立性，如果比较简单，可以通过等式约束，将设计变量的维数减小，从而消除了等式约束的要求。

或者，把每一个等式约束化为两个不等式约束：

$$h_i(\mathbf{x})\geqslant 0 \qquad -h_i(\mathbf{x})\geqslant 0 \qquad i=1,\ 2,\cdots,\ m$$

甚至在很多具体问题上，常常可以把等式约束改成稍稍改型的不等式约束，总是设法把约束条件只有不等式约束条件(2-18)的形式。

在无约束情况下的目标函数，设计变量 x 可取值的范围 D，在附加约束条件以后，能取的部分还要缩小为 D_1，我们称

$$\mathbf{x}\in D_1\subset D\subset R^n$$

的值是问题的“可行解”，D_1 称为“可行域”。最优化问题的目的是求出目标函数的最优可行解。有的时候，我们把问题的可行解称为“次最优解”，最优可行解简称为“最优解”。最优解和次

最优解都是我们探讨的对象，因为有些非常复杂的问题，最优解很难找，这时，即使有最优解存在，我们还是对于次最优解感到很有兴趣，因为它在有关的另一类情况下可能就是最优解。所以，在伸缩约束条件的多少的变动以后，可以对所有可能出现的相对最优的次最优解，探讨它们的灵敏性问题。

总之，最优化问题提出了设计变量、目标函数、约束条件三个方面，而且这三个方面又是一个整体，在具体着手设计，寻找最优设计的效果时，须要同时考虑。为了说明这种设计问题的数学模型的建立，我们在这里将介绍船舶主尺度的选择，杆件构架，螺旋桨设计和汽轮机设计四个问题，简要的说明一下最优化问题的提出和解决的过程。

§2 船舶主尺度的选择

船舶的主尺度是一些设计变量，船长L，型宽B，型深H，吃水T和方型系数C_B，在给定的船舶载重量，货仓容积，航速，续航力，主尺度限制等约束条件下，选择船舶的最适宜的主尺度，使目标函数（例如经济成本、航速）达最优值或次最优值。这就是一种十分典型的最优化问题。

过去，选择船舶主尺度的方法，多是采用“网格法”，这种方法根据经验，对要待定的设计变量L，B，H，T，C_B等参数给以适当的某些数值，经过各参数值的不同组合，构成许多种方案，对每个方案逐个计算，比较，求得较优的解案。

这种网格法的解法，相当于在可行域D_1中用一些各参数是常数的超平面，将n维（$n=5$，或$n=6$等）实空间R^n的子空间D_1划分成一个个小的超立方体，各个顶点就给出我们要设计的不同的方案。

网格法又称为“格栅法”，它的特点是能够找出整体的最优解，但是计算复杂，没有引进近代数学上的一些技巧。

我们现在的办法是把问题归结为一种带约束的非线性规划问

题，非线性规划是目前研究得比较深入的一种数学问题，利用这种数学方法，我们就可以找出最优设计要求的解。

有了最优设计的一套数学方法之后，主要的问题是如何正确的选用适当个数的设计变量，化简约束条件，确定目标函数的数学模型。这些都有赖于有设计经验的工程人员根据具体情况来解决。

例如，在船舶主尺度的最优设计中，如果我们要设计的船舶是某种型号的驱逐舰，我们就要以“航行速度”做为目标函数，最优化的目的就是选择主要的一些主尺度使目标函数达极大值；如果我们设计的是针对航行在特定选好的长途或短途往返的航线上的商船，我们可以选“经济成本”为目标函数，然后选主尺度变量使目标函数达极小值。

下面简要介绍一下以“吨成本”作为目标函数，用最优化方法选择船舶的主尺度。

（1）设计变量：

若已知设计吃水是$T=6.5$米，并且已经选定了主机类型。

以方型系数C_B、船长L、型宽B、型深H四个主尺度，外加载重量DWT、速率V_S这六个参数做为设计变量，也就是设计变量是在R^6中的某个子集D中取：

$$\mathbf{x}=(C_B,\ L,\ B,\ H,\ DWT,\ V_S)^T$$

若$\xi_1=C_B$，$\xi_2=L$，$\xi_3=B$，$\xi_4=H$，$\xi_5=DWT$，$\xi_6=V_S$，则

$$\mathbf{x}=(\xi_1,\ \xi_2,\ \xi_3,\ \xi_4,\ \xi_5,\ \xi_6)^T$$

并且

$$\mathbf{x}\in D\subset R^6$$

（2）目标函数：

以经济指标“吨成本”为目标函数，即

吨成本＝年度的总成本（总支出）/年货运量（每年吨）

年度的总成本包括：

1）造价：包括船的造价，如钢质船体、木制舾装、机电设

备造价和其它间接有关费用，如辅助工程费、设计和验船费、企业管理费、利润、税金等。

2）折旧维修费：包括折旧、保险与维修费用等。

3）船员工资：包括船员工资、伙食津贴、航行津贴等。

4）港口费：包括按年航次数、船吨、载货量计算的港口费及货物装卸费等。

5）燃料：包括按航行、停泊、年航次数等计算燃料、润滑料费用等。

（3）约束条件：

在设计任务中提出的载货量、容积、主尺度限制，航速要求以及与问题有关的物理量等都可作为约束条件。例如，由航速及给定功率之间的关系；由舱容和尺度之间的关系；由浮力方程和重量方程可以建立四个等式约束；由稳性关系及由设计变量取值范围等也可建立一些不等式约束。

交通部上海船舶运输科学研究所曾用这个方法对一艘5000吨干货船进行最优设计，按以上选择的方案，使用非线性规划的数学方法对吃水 $T=6.5$ 米的情况给出4个等式约束，8个不等式约束，以“吨成本”为目标函数，求出的最优解是：

$$\mathbf{x}^*=(0.695,\ 98.0,\ 16.0,\ 9.0,\ 5010.3,\ 13.3)^T$$

$$F(\mathbf{x}^*)=25.45$$

§3 杆架结构设计模型

在结构设计中，有很多类最优化问题，例如材料和外形的选择，某一种给定的几何结构构件的最优化问题等等。

一个对称三杆平面构架的构件最优化问题是设计一种给定几何形状的三杆构架，使它在合于结构免于受损的约束条件下，总重量最小。

如图2-2所示三杆构架，设杆长给定，设计参数是三个杆件的横截面积 A_1，A_2，A_3。在结构平面对称时，$A_1=A_3$，所以就

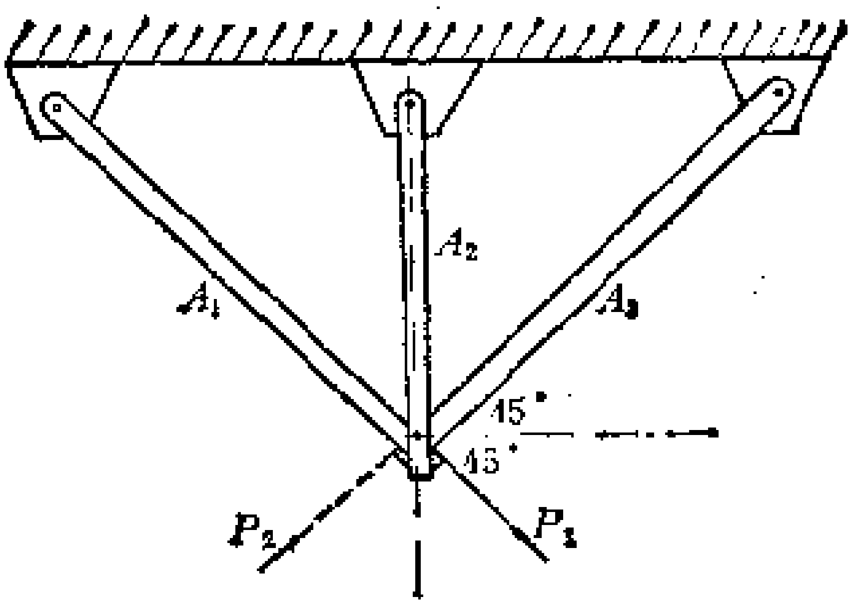


图2-2 三杆构架

仅有两个设计变量 A_1 和 A_2。

这个构架的加载条件是图中所示的 P_1 或 P_2，但不是同时加载。它们的大小相等，等于 P，并且是对称的作用。所以，极小化的目标函数是：

$$W=\sum_{i=1}^{3}\rho_i L_i A_i=2\rho_1 L_1 A_1+\rho_2 L_2 A_2 \qquad (2\text{-}19)$$

其中：ρ_i 和 L_i 各是第 i 个杆件的密度和长度。

很多结构设计问题是针对容许应力这个临界强度极限来考虑的。一般情况是设计工作者自由校正结构的设计变量，使约束得到满足，所以设计的过程就是修改合于约束条件的可实现的可行设计变量，以极小化某一种反映损耗的目标函数，使损耗费用最小。

简单的构架结构的约束条件，包含各个杆件应力的上界和杆件横截面积的下界，在这些约束条件下，常常可以得到所有的设计变量都能够满足的设计。特别是，当设计变量仅由应力约束控制时，就是通常称谓的"满应力设计"，如果既有应力约束又有面积度量的约束，则称它们是一种"满约束设计"。

在图 2-2 的三杆杆系中，如果用 F_i 表示第 i 个杆件的内力，则在 P_1 加载的情况下，用力学方法解此超静定结构，得各杆件内力如下：

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \frac{P_1}{\sqrt{2}A_1A_3 + A_1A_2 + A_2A_3} \begin{pmatrix} A_1A_2 + \sqrt{2}A_1A_3 \\ \sqrt{2}A_2A_3 \\ -A_2A_3 \end{pmatrix}$$

也就是：

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \frac{P_1}{\sqrt{2}A_1 + 2A_2} \begin{pmatrix} A_2 + \sqrt{2}A_1 \\ \sqrt{2}A_2 \\ -A_2 \end{pmatrix} \tag{2-20}$$

类似地，关于 P_2 就有：

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \frac{P_2}{\sqrt{2}A_1 + 2A_2} \begin{pmatrix} -A_2 \\ \sqrt{2}A_2 \\ A_2 + \sqrt{2}A_1 \end{pmatrix} \tag{2-21}$$

以式（2-20）的情况讨论，各杆件应力表示是：

$$\begin{pmatrix} \dfrac{F_1}{A_1} \\ \dfrac{F_2}{A_2} \\ \dfrac{F_3}{A_3} \end{pmatrix} = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = \frac{P_1}{\sqrt{2}A_1^2 + 2A_1A_2} \begin{pmatrix} A_2 + \sqrt{2}A_1 \\ \sqrt{2}A_1 \\ -A_2 \end{pmatrix}$$

要它们合于满应力设计的要求，就需要有：

$$|\sigma_i| \leqslant |[\sigma_i]|$$

若已知各杆件的材料相同，并且密度 $\rho = 0.1$；受拉的容许应力 $[\sigma_+] = 20000$，受压的容许应力是 $[\sigma_-] = -15000$；$P = 20000$，则应有：

$$\frac{P_1}{\sqrt{2}A_1^2 + 2A_1A_2} \begin{pmatrix} A_2 + \sqrt{2}A_1 \\ \sqrt{2}A_1 \\ |-A_2| \end{pmatrix} \leqslant \begin{pmatrix} [\sigma_+] \\ [\sigma_+] \\ |[\sigma_-]| \end{pmatrix}$$

也就是：

$$2A_2\left(1-\frac{1}{2A_1}\right)\geqslant\sqrt{2}(1-A_1)$$

$$A_2\geqslant\frac{\sqrt{2}(1-A_1)}{2} \tag{2-22}$$

$$3\sqrt{2}A_1^2\geqslant\pm A_2(4-6A_1)$$

式（2-22）就是现在满应力设计的约束条件。

同时，目标函数（1-19）在现在的情况下是：

$$W=W(A_1,A_2)=0.1\times10\sqrt{2}\times2A_1+0.1\times10A_2$$

$$=2\sqrt{2}A_1+A_2 \tag{2-23}$$

解决这种带约束（2-22）使式（2-23）极小的选出 A_1，A_2 解的最优化问题，工程人员已习惯于使用"应力比"即"比例满应力"的方法求解，在第三章中，我们还将讨论这个问题的解法。

关于三杆构件的"满约束设计"，如果把目标函数式（2-19）写成：

$$F(x)=a_1x_1+a_2x_2+a_3x_3 \tag{2-24}$$

则 $a_i=\rho_iL_i$，$x_i=A_i$（$i=1, 2, 3$），度量约束是：

$$x_{\min}\leqslant x_j\leqslant x_{\max} \qquad j=1, 2, 3 \tag{2-25}$$

应力约束是：

$$\sigma_{\min}\leqslant\bar{\sigma}_{rs}\leqslant\sigma_{\max} \qquad r=1, 2, 3 \tag{2-26}$$

$$s=1, 2$$

这里的 S 表示图 2-3 中两种加载的情况，r 表示第 r 个杆件，所以 $\bar{\sigma}_{rs}$ 表示加载 S 的情况下第 r 个杆件的应力。

满约束设计式（2-24）、（2-25）、（2-26）可以通过第三章介绍的序列无约束极小化技巧（简称为 SUMT 方法），化为一系列如下的无约束最优化问题：

极小化 $\quad F(x)+r^{(k)}[b_1(x)+b_2(x)]$ $\qquad$ (2-27)

$$r^{(k)}>0 \qquad r^{(k+1)}=cr^{(k)} \qquad 0<c<1$$

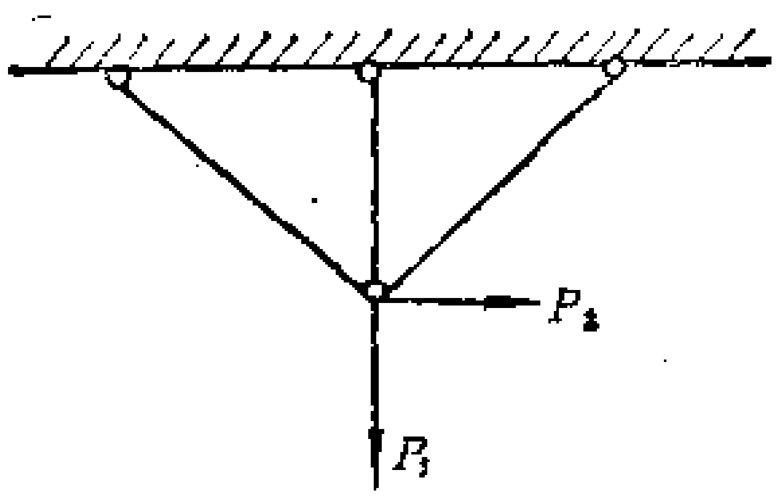


图2-3　三杆构架的满约束设计

其中

$$b_1(x)=\sum_{j=1}^{3}\left(\frac{1}{x_{max}-x_j}+\frac{1}{x_j-x_{min}}\right)(x_{max}-x_{min})$$

$$b_2(x)=\sum_{r=1}^{3}\sum_{s=1}^{2}\left(\frac{1}{\sigma_{max}-\bar{\sigma}_{rs}}+\frac{1}{\bar{\sigma}_{rs}-\sigma_{min}}\right)(\sigma_{max}-x_{min})$$

这种问题就是一般最优化问题，即

$$\text{极小化}\quad F(x)$$

$$\text{使合于}\quad g_i(x)\geqslant 0\qquad i=1,\ 2,\cdots,\ P \tag{2-28}$$

的又一种典型例子。

§4　船舶可调螺距螺旋桨的设计模型

设计螺旋桨常用的方法有两种，一种是以螺旋桨环流理论为基础的理论设计方法；另一种是图谱设计方法。图谱设计方法简单易行，目前船厂和设计部门广泛使用。

螺旋桨的各种设计图谱是根据模型桨系列的“敞水”试验结果绘制成功的。过去，绘制图谱是由设计人员用手工方法，以表格的方式来计算，这是一种非常繁重的工作。近年来，随着电子计算机的广泛应用，各种设计图谱也按手工计算的步骤编写了一些程序，代替了手工方法。如果再采用最优设计方法，就可以更好地发挥电子计算机的作用，这时，除了使设计变量满足设计要求以外，还要在某种意义下达到最优的性能指标，从而获得

更好的综合结果。

采用最优设计方法，首先要建立一个合理的数学模型，它包括一个目标函数和一些约束条件。但是，螺旋桨的设计本来就是一个比较复杂的问题，例如，和螺旋桨推进性能有关的因素就有三、四十个。而可调螺距的螺旋桨设计还要考虑在不同的工况下，对推进效率、空泡性能、振动、强度等各个方面作进一步考核，并且在选取影响性能的螺旋桨要素时，使矛盾的各个方面取得统一，例如，对于渔轮，拖船等船舶，航速和功率不甚大，考虑空泡性能时，要在避免发生空泡的情况下尽量求得最高的推进效率，并进行强度校核，同时针对船舶的使用要求，确定设计螺距比，检验在不同工况时是否都能达到设计要求，从而确定螺旋桨的主要尺寸。

下面，简单讨论一下螺旋桨的各个参数对性能的影响，从而找出哪些是主要的，哪些是次要的，进而作为设计依据。

这里所讨论的设计问题是指所谓“终结设计”，也就是说对新船选用现成的标准主机，对旧船是选配螺旋桨的问题。因此，主机的功率、转速都是给定的，螺旋桨的个数也由其它方面加以决定，这些参数就无须再在设计中考虑。

另外，当前普遍采用的是使用系列图谱设计，因此，桨毂的形状及桨叶轮廓、剖面形状、厚度分布、纵倾及侧斜等一些几何尺寸都是规定好的，在设计中可以不考虑它们。

大家知道，螺旋桨的推进效率是主要的，必须在高效率的情况下，确定螺旋桨的主要尺寸和形状特征。例如，螺旋桨的直径和螺距比对螺旋桨水动力特性有很大影响，当转速一定时，有一个最优直径，它使螺旋桨的敞水效率最好；同时可以找到对应的螺距比。离开这一最优的位置，螺旋桨的效率是比较低的。当然，螺旋桨的直径等设计尺寸还要受到船舶全部结构及吃水深度等方面的限制。因此，设计的关键在于满足这些条件的情况下，如何首先解决螺旋桨的主要尺寸和形状参数的“最优化”。

另外，为了避免空泡发生，需要选用较大的盘面比，但是盘面比选得太大，会使效率降低，因此，应该根据空泡校核曲线选择不发生空泡的最小盘面比。还有振动问题，习惯上是在桨叶数选择时来考虑；而在强度方面，一般则作为一种事后校验计算。

根据上面的讨论，设计变量选取下列四个：

1）螺旋桨直径D；

2）螺距比H/D；

3）盘面比 A/A_d；

4）可以达到的航速 V_t，或者可以达到的拖曳力 T_0。

在这个基础上，来建立目标函数，等约束条件和不等约束条件。

（1）目标函数：

在螺旋桨设计中，总是希望整个推进系统的效率最高，也就是说在一定的功率下，要求船速尽可能高，或者在一定的拖速下，拖曳力尽可能的大，因此，就选取推进系数作指标，要使推进系数 P_c(＜1）达到最大，也就是取目标函数为 1 减去 $C_p(1.0-C_p)$ 使它达到最小。这里：

$$C_p = \eta_0 \cdot \eta_R \cdot \eta_H \cdot \eta_S \tag{2-29}$$

式中 η_0——螺旋桨敞水系数；

η_R——相对旋转效率；

η_H——船身效率；

η_S——传送效率。

（2）等式约束条件：

螺旋桨的设计是使主机、船体、螺旋桨三者协调工作，达到两个平衡：一个是主机发出的功率（考虑到轴系损失）使螺旋桨必须全部吸收，即主机提供给螺旋桨的扭矩要等于螺旋桨吸收的扭矩，此时主机才能平稳的工作；另一个是螺旋桨发出的有效推力要等于船体在对应航速时的阻力及拖曳力二者之和。这时，船速才能以均匀速度前进。下面根据为设计所提供的阻力与推进资

料来建立上述两个平衡关系式作为等式约束条件。

阻力与推进试验可提供如下一些资料：

1）船速与有效马力之间的对应关系，表现了船型阻力特性；

2）伴流分数、推力减额及相对旋转效率；

3）主机功率、主机转速及传送效率；

4）螺旋桨系列的敞水性能。

设船舶以速度 V_s 前进，主有效推力 T_e。

因 $$\mathrm{EHP}=T_e(0.5144V_s)/75\times T_s$$

所以 $$T_e=\mathrm{EHP}\times 75/0.5144J_s\times V_s(\text{公斤})$$

式中 EHP——对应于 V_s 的有效马力，马力；

J_s——螺旋桨个数。

考虑到船身影响，螺旋桨应该提供的推力 T_{e_0} 为：

$$T_{e_0}=\frac{T_e}{1-t}$$

式中 t——推力减额。

再设船舶的拖曳力为 T_0，则螺旋桨所提供的总推力用推力系数 K_{T_1} 表示应为：

$$K_{T_1}=\frac{T_{e_0}+T_0}{\rho n^2D^4}$$

式中 n——螺旋桨每秒钟转速，转/秒；

D——螺旋桨直径，米；

ρ——水的密度。

根据螺旋桨敞水试验结果，得到推力系数 K_T 与螺旋桨的几何参数盘面比 A/A_d、螺距比 H/D 及运动参数进速系数 J_p 的关系曲线，再用回归分析对推力系数进行处理，不难得到推力系数多项式为：

$$K_T=\sum_i\sum_j\sum_k B_{T_{ijk}}(A/A_d)^i\cdot(H/D)^j(J_p)^k$$

式中 $$J_p=\frac{V_a}{n\cdot D}=\frac{0.5144(1-\omega)V_s}{n\cdot D}$$

因此，第一个等式约束条件——推力平衡为：

$$K_{T_1}-K_T=0 \tag{2-30}$$

现在来看第二个等式约束：

设螺旋桨收到的功率（马力）为：

$$\mathrm{DHP}=\eta_s\cdot N_e=2\pi\cdot M\cdot n/75.0$$

式中 M——机器提供的螺旋桨的扭矩，

$$M=\frac{75\cdot\eta_s\cdot N_e}{2\pi\cdot n}$$

η_s——传送效率；

N_e——机器的功率。

将M化为扭矩系数 K_{Q_1}，则：

$$K_{Q_1}=\frac{75\cdot\eta_s\cdot N_e}{2\pi\cdot n\cdot p\cdot n^2D^5}=\frac{75\cdot\eta_s N_e}{2\pi\cdot p\cdot n^3\cdot D^5}$$

同样，根据螺旋桨敞水系列试验，得到扭矩系数 K_Q 与 A/A_d，H/D，J_p 之间的关系曲线，再用回归分析方法得到多项式拟合方法为：

$$K_Q=\sum_i\sum_j\sum_k B_{Q_{ijk}}(A/A_d)^i(H/D)^j(J_p)^k$$

因此，第二个等式约束条件——扭矩平衡为：

$$K_{Q_1}-K_Q=0 \tag{2-31}$$

（3）不等式约束：

首先，考虑空泡性能问题。主要考虑不发生空泡的最小盘面比。也就是，在不发生空泡的前提下，求得最高的螺旋桨效率。柏利尔（Burrill）的空泡校验曲线已经化为对数形式的公式，便于电子计算机应用。设 $(A/A_d)_0$为空泡校验时，不发生空泡的最小盘面比，它与许多因素有关为：

$$(A/A_d)_0=f(V_a,\ n,\ D,\ H_a,\ H_w,\ t,\ H/D,\ T\cdots)$$

式中 V_a——螺旋桨进速；

n——螺旋桨转速；

D——螺旋桨直径；

H_s——轴沉深；

H_z——吃水；

t——水温；

D/H——螺距比；

T——螺旋桨的推力。

因而，建立第一个不等式约束——盘面比 A/A_d 应大于这不发生空泡的最小盘面比 $(A/A_d)_0$，即

$$A/A_d-(A/A_d)_0 \geqslant 0 \tag{2-32}$$

其它不等式约束为边界约束或坐标约束，如螺旋桨的敞水效率可表示为：

$$\eta_0=\frac{J_p}{2\pi}\cdot\frac{K_T}{K_Q} \tag{2-33}$$

$$0\leqslant\eta_0\leqslant 1$$

螺旋桨推进效率为：

$$\eta_p=\eta_0\cdot\eta_r\cdot\eta_H \tag{2-34}$$

$$0\leqslant\eta_p\leqslant 1$$

式中　η_r——相对旋转效率，由设计人员根据不同情况合理选取；

η_H——船身效率，$\eta_H=\dfrac{(1-t)}{(1-\omega)}$。

螺距比：

$$(H/D)_{min}\leqslant(H/D)\leqslant(H/D)_{max} \tag{2-35}$$

盘面比：

$$(A/A_d)_{min}\leqslant(A/A_d)\leqslant(A/A_d)_{max} \tag{2-36}$$

螺旋桨直径：

$$D_{min}\leqslant D\leqslant D_{max} \tag{2-37}$$

航速：

$$(V_s)_{min}\leqslant V_s\leqslant(V_s)_{max} \tag{2-38}$$

拖曳力：

$$0.4(T_0)_g = T_0 \leqslant 1.2(T_0)_g \tag{2-39}$$

式中 $(T_0)_g$——给定拖曳力。

根据上述分析，使用上海交通大学船模试验池最近完成的JDC-3-50系列螺旋桨模型试验资料，对某拖轮进行了螺旋桨的优化设计计算，现介绍如下：

已知：主机功率：4500马力；

主机转速：200 转/分；

传送效率：$\eta_r = 0.96$；

相对旋转效率：$\eta_R = 0.93$；

伴流系数：

自由航行时 $\omega_1 = 0.131$；

拖曳航行时 $\omega_2 = 0.097$；

推力减额系数：

自由航行时 $t_1 = 0.131$；

拖曳航行时 $t_2 = 0.160$；

拖曳力：$T_0 = 35.0$吨；

轴线沉没深度：$H = 3.1$ 米；

叶数：$Z = 3$。

经过伸缩保差法直接优化计算结果得：

直径：$D = 3.38$ 米；

盘面积比：$A/A_d = 0.601$；

螺旋桨敞水效率：（自由航行时）$\eta_0 = 0.683$；

（拖曳航行时）$\eta_0 = 0.570$；

设计螺距比：（自由航行时）$H/D = 0.835$；

（拖曳航行时）$H/D = 0.757$；

达到航速：（自由航行时）$V_s = 19.4$ 节；

（拖曳力为35吨航行时）$V_s = 13.2$ 节；

推进系数：（自由航行时）$C_p = 0.635$；

、　（拖曳航行时）$C_p=0.493$。

§5　汽轮机设计模型

蒸汽动力装置部件的一种最普通的组合就是所谓兰金(Rankine)循环，见图2-4。在这个循环中，重要的设计部分是汽轮和冷凝器的两个部分。

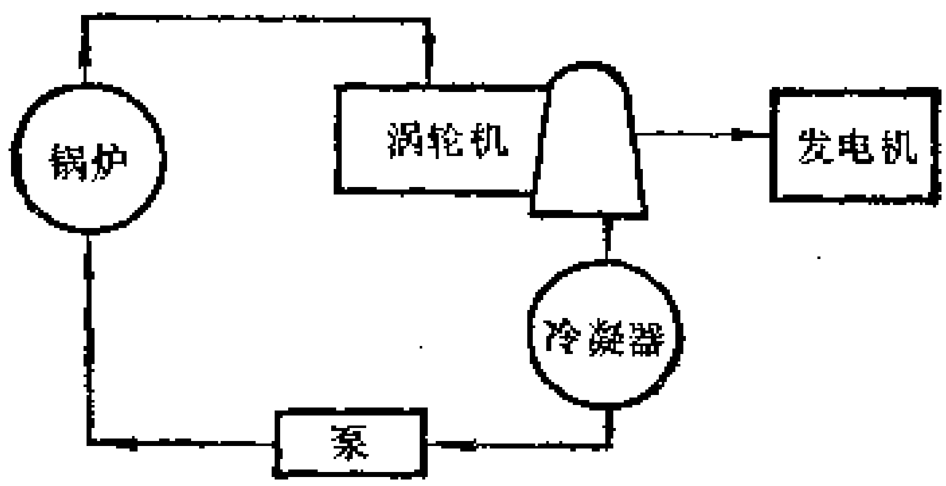


图2-4　兰金循环示意图

关于由汽轮发电机与冷凝器构成的这样一个冷凝系统的最优设计问题，我们要极小化这个冷凝系统的年消耗C(例如每年付费用C元)，这种消费的计算包括汽轮机运转消耗$C_{O,T}$，固定费用$C_{F,T}$，冷凝器的运转消耗$C_{O,C}$以及冷凝器的固定费用$C_{F,C}$，所以整个费用消耗的经济指标给出待最优化的目标函数为：

$$C=C_{O,T}+C_{O,C}+C_{F,T}+C_{F,C} \tag{2-40}$$

一般情况，每一消耗项都可以用经验的或理论分析给出的幂函数形式表示：

$$C_i t_1^{a_{i1}} t_2^{a_{i2}} \cdots\cdots t_m^{a_{im}} \tag{2-41}$$

式中：C_i是正的常数，t_1，t_2，……t_m是正的设计变量，指数a_{ij}是任意的实数常数。

如果我们在导出有关冷凝系统的约束条件又都是（2-41）形式的某些不等式的约束条件，那么这种最优化问题就容易用第三章介绍的几何规划来求解。

这就是1976年艾克（Ecker）和威金（Wiebking）进行冷凝系统最优设计的出发点。

给出约束条件的意思，是说明在目标函数（2-40）中的设计变量不能任意选取，必须满足一定的设计关系和设计要求。这些约束关系，有的是等式的约束形式，有的是不等式约束形式。实际上，我们总可以保持只有不等式的约束关系。

譬如：在一个热交换器中，从热到冷流体温度下降的平均数值，将等于输送管道中，管边界内部和外部的温度平均下降量之和，似乎应该写出等式约束：

$$\Delta T_m = \Delta T_{mo} + \Delta T_{mi}$$

但是，如果考虑到其它的温度降落 ΔT，就可以写成不等式的约束形式：

$$\Delta T_m \geqslant \Delta T_{mo} + \Delta T_{mi} \tag{2-42}$$

就目标函数而言，汽轮机的固定费用 $C_{F,T}$ 是最后叶片环面积 A_{an} 的一个函数，即

$$C_{F,T} = p_a c_{an} A_{an}^{b_{an}} \tag{2-43}$$

式中：c_{an} 是环面积消耗关系的系数；p_a 是估值减少率系数。

汽轮机的运转消耗包括燃料消耗和交换能量的消耗，在全年满负荷运转的设定情况下，若加入锅炉或核反应堆的热量是常量，那么最优化的模型，消耗就可以降低；交换能量的损耗常常是在夏天，冷却水的温度是比较高的。一个具有最优排气环与冷凝器的汽轮机将能给受相对于额定的发电容量 kW_R 的消耗，失去的容量必须由电力公司按每千瓦递增的费用代替，从而交换能量的耗费是：

$$C_{O,T} = c_e t (kW_R - kV) \tag{2-44}$$

式中：c_e 是每小时千瓦的费用；t 是以小时计的有效运转时间。

类似的问题是在低冷却水温度时，相对于额定条件，就有发电容量的一个收益，从而减低了运转消耗，所以式（2-44）可以用

供应的环流水量的形式写成：

$$C_{O,T}=c_e t\eta_G(q-q_K)/3412\cdot 75$$

如把常数略去，就是：

$$C_{O,T}=c_e t\eta_G q/3412\cdot 75 \qquad (2\text{-}45)$$

式中：η_G是发电效率。

另外，冷凝器的固定费用设定正比于冷凝管曲面积A_0，所以：

$$C_{F,C}=p_e c_t A_0 \qquad (2\text{-}46)$$

而冷凝器的运转消耗就是把水泵到冷凝管去的费用，它是：

$$C_{O,C}=\frac{c_e t W\Delta P}{\rho\eta_p} \qquad (2\text{-}47)$$

式（2-46）中的c_t表示冷凝器曲面单位造价；式（2-47）中的W是水流速率，ρ是密度，ΔP是内管的全部压降，η_p是泵的效率。

热传递面积是：

$$A_0=\pi ND_0L \qquad (2\text{-}48)$$

式中：D_0表示外管直径；L表示管长；N表示冷凝器中的管子数。

这时，冷凝管的全部压降为：

$$\Delta P=B\Delta P_s \qquad (2\text{-}49)$$

式中：B是压降因子。

而经过直管的压降是：

$$\Delta P_s=\frac{32fLW^2}{\pi^2\rho gD_i^5N^2} \qquad (2\text{-}50)$$

式中：D_i是内管直径；f是阻力因子，$f=\frac{0.046}{(\mathrm{Re})^{0.2}}$，其中$\mathrm{Re}=\frac{4W}{\pi\mu D_iN}$，这里$\mu$是粘度。

所以，把式（2-48）、（2-49）和（2-55）代入式（2-46）和（2-47）就有：

$$C_{F,C}=c_3D_0LN \qquad (2\text{-}51)$$

和

$$C_{O,C}=\frac{c_4 L}{D_i^{4.8}N^{1.8}} \tag{2-52}$$

最后，得到目标函数（2-40）的形式是：

$$g_0\equiv c_1A_{an}^{ban}+c_2q+c_3ND_0L+c_4\frac{L}{D_i^{4.8}N^{1.8}} \tag{2-53}$$

式中

$$c_1=c_{an}p_e$$

$$c_2=\frac{c_e t\eta_G}{3412\cdot75}$$

$$c_3=\pi c_t p_e$$

$$c_4=\frac{320.046\,c_e BtW^{2.8}\left(\frac{\mu}{4}\right)^{0.2}}{7383600^2 g\eta_p\rho^2\pi^{1.8}}$$

关于约束条件，经过分析，写成能用几何规划求解的形式，包括下列十二个：

$$g_1\equiv c_5\frac{Q_a}{q}+c_6\frac{Q_aT_s}{q}+c_7\frac{Q_af_M\Delta h'}{q}\leqslant 1$$

式中

$$c_5=h_{2,R}+a_1f_{M,R}+0.87a_4a_6-a_2$$

$$c_6=b_1f_{M,R}+0.87b_4a_6-b_2$$

$$c_7=1$$

Q_a：冷凝器每小时流量；

T_s：蒸汽的饱和温度；

f_M：在膨胀线末点的温度改正因子；

Δh：排气损失，$\Delta h'=\Delta h-a_6$；

h_2：膨胀线末点单位量的热量Btu●。

$$g_2\equiv c_8\frac{1}{f_M}+c_9\frac{T_s}{f_M}\leqslant 1$$

式中

$$c_8=0.87a_4$$

$$c_9=0.87b_4$$

● Btu为英国热量单位（=252卡）。

$$g_3 \equiv c_{10}\frac{T_s^{2b_t}}{A_{an}^2\Delta h'} + c_{11}\frac{A_{an}}{T_s^{b_t}\Delta h'} \leqslant 1$$

式中

$$c_{10} = b_{61}(a_5 Q_{a,R})^k$$

$$c_{11} = \frac{b_{62}}{Q_{a,R}a_5}$$

$$g_4 \equiv c_{12}\frac{1}{T_s} + c_{13}\frac{q}{T_s} + c_{14}\frac{q^{4/3}}{T_s N^{7/6} D_o L^{4/3}}$$

$$+ c_{15}\frac{qD_i^{0.8}}{T_s L N^{0.2}} \leqslant 1$$

式中

$$c_{13} = \frac{1}{2c_p W}$$

$$c_{14} = \left(\frac{1}{\pi 0.725}\right)^{4/3}\left[\frac{\mu_f}{23600^2 k_f^3 \rho_f^2 g\lambda}\right]^{1/3}$$

$$c_{15} = (0.023\pi k P_r^{0.4})^{-1}\left[\frac{4W}{\pi\mu}\right]^{-0.8}$$

c_p：比热；

λ：气化潜热；

k：热导性；

P_r：普朗特（Prandtl）数。

$$g_5 \equiv c_{16}\frac{1}{Q_s} + c_{17}T_s \leqslant 1$$

式中

$$c_{16} = \frac{Q_{a,R}}{a_5}$$

$$c_{17} = -\frac{b_3}{a_3}$$

$$g_6 \equiv c_{18}\frac{1}{D_o} + c_{19}\frac{D_i}{D_o} \leqslant 1$$

式中

$$c_{18}=2$$

$$c_{19}=1$$

$$g_7\equiv c_{20}\frac{1}{D_o}\leqslant 1$$

式中

$$c_{20}=D_{o,\min}$$

$$g_8\equiv c_{21}\frac{T_s^{b_s}}{A_{an}}\leqslant 1$$

式中

$$c_{21}=a_5\frac{Q_{a,R}}{V_u}$$

$$g_9\equiv c_{22}\frac{A_{an}}{T_s^{b_s}}\leqslant 1$$

式中

$$c_{22}=\frac{V_1}{Q_{a,R}a_5}$$

$$g_{10}\equiv c_{23}\frac{Q_a}{A_{an}}\leqslant 1$$

式中

$c_{23}=\dfrac{1}{\tau_{\max}}$，其中$\tau_{\max}$为容许的最后环叶片负荷。

$$g_{11}\equiv c_{24}ND_i^2\leqslant 1$$

式中

$$c_{24}=\frac{3600u_{\min}\pi\rho}{4W}$$

其中 u 为速度。

$$g_{12}\equiv c_{25}T_n\leqslant 1$$

式中

$c_{25}=\dfrac{1}{T_c+\Delta T_{\max}}$，其中 $\Delta T_{\max}$ 为最大容许的冷水温度增加；

T_n；热水温度；

T_c：冷水温度。

以上，所有加上 R 下脚标的均表示额定（平均）条件；加上 f 下脚标的表示外层。

经过几何规划的最优解，上述冷凝系统的最优值 E.-W. 的结果是：

汽轮机设计变量

排气压力	1.231	英寸水银柱
环面积	409.2	平方英尺
环速	929.4	每秒英尺
最末叶片负荷	6556	每小时平方英尺磅

冷凝器设计变量

废弃热量	2.550×10^{9}	每小时Btu
蒸汽的饱和温度	85.45	华氏度
外曲面面积	0.2787×10^{6}	平方英尺
管子个数	13.660	
水速度	6.517	每秒英尺

第三章　数学规划及其解法

§1　非线性规划

我们已经知道，从实际问题提出的待最优化的设计对象，包括设计变量的选择，目标函数的确立，约束条件的全面分析，才能形成最优化问题的一种数学模型的整体。

一般地说，经过分析，最优化问题可以归结为一个如下的数学规划问题：

$$\left.\begin{array}{l}
\text{对目标函数 } F(\mathbf{x}) \text{ 极小化，即} \\
\qquad \min\limits_{\mathbf{x}} F(\mathbf{x}) \qquad \mathbf{x} \in D \subset R^n \\
\text{使合于 } m \text{ 个等式约束条件，即} \\
\qquad h_i(\mathbf{x}) = 0 \qquad i = 1, 2, \cdots, m \\
\text{并合于 } p-m \text{ 个不等式约束条件，即} \\
\qquad g_i(\mathbf{x}) \geqslant 0 \qquad i = m+1, \cdots, p
\end{array}\right\} \qquad (3\text{-}1)$$

所以可简写为：

$$\min_{\mathbf{x}\in D\subset R^n} \{F(\mathbf{x}) \mid h_i(\mathbf{x}) = 0 \qquad i = 1, 2, \cdots, m,$$
$$g_i(\mathbf{x}) \geqslant 0 \qquad i = m+1, \cdots, p\}$$

经常的情况是 $F(\mathbf{x})$ 是一个 n 向量 $\mathbf{x}$ 的非线性函数，$h_i(\mathbf{x})$ 或 $g_i(\mathbf{x})$ 也不必是 n 向量 $\mathbf{x}$ 的线性函数，所以我们把这样的(3-1)问题称为是一种“非线性规划”。

按（3-1）形式写出的非线性规划是针对使用序列无约束极小化技巧（简写为 SUMT）的解法。在其他的情况，要极大化目标函数时，可以将目标函数乘以负号改用极小化；避免使用等式约束，可以把它们改写成两个不等式约束，或者减少独立设计变量，或者适当地修改为不等式约束；如果从理论上进行探讨，更多地是把 $g_i(\mathbf{x}) \geqslant 0$ 写成 $-g_i(\mathbf{x}) \leqslant 0$ 以归一的分析凸函数问题。

实际上，在给出m个关于n向量x的等式约束以后，n向量x的独立性就将减少m个，使原来可以独立选取的x中的n个分量减少为只能就$n-m$个分量独立选取，其它的分量则由等式约束来确定。另外，不等式约束条件再把已经减少维数的范围缩小，所以，整个非线性规划可以选取的有意义的设计变量范围，总要比目标函数有定义的域D小些，我们称它是一个“可行域”，表为D_1，它是D的一个子集，即

$$D_1 \subset D \subset R^n$$

我们在这里主要的是介绍用SUMT方法来求解非线性规划，这种方法是1961年前后开始应用于化学工业上，1966年费阿柯（Fiacco）和麦克柯米克（McCormick）有效地发展了的。1969年美国导弹研究所给出了SUMT的电子计算机编码。近年来，这种方法在工程设计的各个领域中得到普遍应用。

SUMT解法能够把非线性目标函数F(x)在非线性不等式约束函数$g_i(\mathrm{x})$($i=m+1,\cdots,p$)，线性等式约束函数$h_i(\mathrm{x})$($i=1,2,\cdots,m$)的情况下，给出收敛效果。限制$h_i(\mathrm{x})$是线性的目的，是使$h_i^2(\mathrm{x})$形成一个凸函数。但是，形成凸函数的$h_i(\mathrm{x})$也不必是线性的。

SUMT解法的基本思想是反复求解一序列的无约束的极小化问题，它的一序列目标函数是由非线性规划的目标函数和约束函数组合而成的。通过这一序列无约束目标函数的解，当其解稳定地趋向一个极限，就得到原非线性规划问题的极小点的解。

非线性规划问题，不能撇开约束条件，单纯求解目标函数的无约束极小化解

$$\min_{\mathrm{x}} F(\mathrm{x}) \quad \mathrm{x} \in D \subset R^n$$

来完成，也就是如果

$$\min_{\mathrm{x}} F(\mathrm{x}) \quad \mathrm{x} \in D \subset R^n$$

$$\min_{x} \{F(x) | h_i(x)=0 \quad i=1, 2, \cdots, m$$

$$g_i(x) \geqslant 0 \quad i=m+1, \cdots, p\} \quad x \in D \subset R^n$$

都存在，则前者的值将不大于后者，后者的极小是相对于约束条件的极小。另外，我们对于无约束极小问题有很多求解方法，所以我们可以把一般有约束的极小化问题转化为无约束极小化问题来求解。

SUMT 解法，就是采用一种适当的组合，对于选定的一种单调下降正值序列 $r^{(0)} > r^{(1)} > \cdots > 0$，写出一个包括目标函数，约束函数在内的评价函数，或称为“惩罚函数”的，例如：

$$P(x, r^{(k)}) = F(x) + (r^{(k)})^{-1/2} \sum_{i=1}^{m} h_i^2(x) + r^{(k)} \sum_{i=m+1}^{p} \frac{1}{g_i(x)} \tag{3-2}$$

SUMT 方法的使用，最早是只有不等式约束情况 写出(3-2)的特例形式为：

$$P(x, r^{(k)}) = F(x) + r^{(k)} \sum_{i=m+1}^{p} \frac{1}{g_i(x)} \tag{3-3}$$

具体求解的过程是：先给定 $r^{(0)}$，例如 $r^{(0)}=1$，按

$$P(x, r^{(0)}) = F(x) + r^{(0)} \sum_{i=m+1}^{p} \frac{1}{g_i(x)}$$

求得极小点 $x^{(0)*}$，如果 $x^{(0)*}$ 是使 $g_i(x) > 0$ ($i=m+1, \cdots, p$) 成立的点，从 $x^{(0)*} = x^{(1)}$ 开始，对 $r^{(1)}$ 求函数(3-3)的极小点 $x^{(1)*}$，再检验 $x^{(1)*}$ 是否是使 $g_i(x) > 0$ ($i=m+1, \cdots, p$) 皆成立的点。如此下去，直到 $x^{(n)}$，$x^{(n+1)}$ 的差合于所要求的接近精度，就得到非线性规划极小点 x^* 的近似解，即

$$x^{(n)} \approx x^*$$

对于不等式约束条件是：

$$g_i(x) \geqslant 0 \qquad i=m+1, \cdots, p$$

我们把那些使它满足严格不等式

$$g_i(x)>0 \qquad i=m+1,\cdots,\ p$$

的点，称为“内点”。当然，用内点做初始点求解（3-2)或(3-3)时，它的解未必是一个可行点。在序列求如式（3-2)所示函数的极小点时，由于采用的序列因子 $r^{(k)}$ 的逐渐减小，$(r^{(k)})^{-1/2}$ 的逐渐加大，就必然迫使 $h_i^2(x)$ 趋近于零才能形成惩罚函数的极小。SUMT 方法的本身，采用内点方法的思想是在式（3-2)和(3-3)中，一旦计算过程的点使不等约束任何一个接近于零时，将从 $r^{(k)}$ 的趋小限制，它只能增加才能使 $1/g_i(x)$ 减小，所以我们说式（3-2）和（3-3）中的

$$\sum_{i=m+1}^{p}\frac{1}{g_i(x)}$$

各项对计算过程起了一种“障碍”作用，并且在 $r^{(k)}$ 非常趋小时，障碍的效应也减小，从而所得序列极小点 $x^{(*)}$ 可以移向某一不等式约束的边界。

SUMT 保持内点的另一种“障碍”项的取法是：

$$\sum_{i=m+1}^{p}\ln\frac{1}{g_i(x)}=-\sum_{i=m+1}^{p}\ln[g_i(x)]$$

所以，惩罚函数也可写成：

$$P(x,\ r^{(k)})=F(x)+(r^{(k)})^{-1}\sum_{i=1}^{m}h_i^2(x)-r^{(k)}\sum_{i=m+1}^{p}\ln g_i(x) \tag{3-4}$$

这时，关于等式约束“消耗”作用可以步调相应于对数减慢的作用而减慢一些。无论是式（3-2）或式（3-4)，等式约束项

$$\sum_{i=1}^{m}h_i^2(x)$$

都是要使它逐渐趋小的“消耗”下去。

事实上，我们可以看出式（3-2）和（3-4）的惩罚函数形式

是更为一般的惩罚函数

$$P(\mathbf{x},\ \rho^{(k)})=F(\mathbf{x})+\sum_{i=1}^{m}\rho_i^{(k)}H(h_i(\mathbf{x}))$$

$$+\sum_{i=m+1}^{p}\rho_i^{(k)}G(g_i(\mathbf{x})) \tag{3-5}$$

的二种特殊形式。

另外，我们可以把所有等式约束改为不等式约束，也就是：

$$h_i(\mathbf{x})=0 \Longrightarrow \begin{array}{l} h_i(\mathbf{x})\geqslant 0 \\ -h_i(\mathbf{x})\geqslant 0 \end{array} \qquad i=1,\ 2,\ \cdots,\ m$$

但是，这将增加了计算的复杂性；同时，我们也可以把不等式约束，附加使用松弛变量 v_i，写成等式约束，也就是：

$$g_i(\mathbf{x})\geqslant 0 \Longrightarrow g_i(\mathbf{x})-v_i^2=0 \qquad i=m+1,\ \cdots,\ p$$

并且由此把非线性规划（2-1）的惩罚函数写成确定的拉格朗日(Lagrange）函数为：

$$P(\mathbf{x},\ \omega)=F(\mathbf{x})+\sum_{i=1}^{m}\omega_i h_i(\mathbf{x})+\sum_{i=m+1}^{p}\omega_i(g_i(\mathbf{x})-v_i^2) \tag{3-6}$$

这时，待定的变量大量增加，它们包括了三大类 x，ω，v，并且它能成立求解的条件比较 SUMT 的方法要苛刻得多。

按上述惩罚函数表出的 SUMT 内点法形式（3-2）、(3-3)、(3-4）以及只有不等式约束的形式

$$P(\mathbf{x},\ r^{(k)})=F(\mathbf{x})-r^{(k)}\sum_{i=m+1}^{p}\ln g_i(\mathbf{x}) \tag{3-7}$$

开始计算时，要选定一个内点 $\mathbf{x}^{(0)}$，序列因子 $r^{(0)}$有很多选法，通常取 $r^{(0)}=1$，然后以 1／4 的比值按几何级数减小，也就是取 $\{r^{(k)}\}$的序列是：

$$1,\ 1/4,\ 1/16,\ 1/64,\ 1/256,\cdots$$

当然，也可以根据具体问题的情况来确定$r^{(k)}\to 0$的序列$\{r^{(k)}\}$的选法。在$r^{(0)}$确定以后，可以极小化$P(\mathbf{x}, r^{(0)})$得到$\mathbf{x}^{(1)}$，如果它是内点，再用$r^{(1)}$极小化$P(\mathbf{x}, r^{(1)})$得$\mathbf{x}^{(2)}$，如此下去，就得到一序列内点$\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(k)}, \cdots$。在一定条件下，在给定的容许误差范围内，就可以得到稳定解$\mathbf{x}^*$的近似值。下面举例说明。

例 1　极小化二元函数

$$F(x_1, x_2)=(x_1-1)^2+(x_2-1)^2$$

$$\text{使合于}\quad g_1(x_1, x_2)=-x_1\leqslant 0$$

$$g_2(x_1, x_2)=-x_2\leqslant 0$$

$$g_3(x_1, x_2)=x_1^2+x_2^2-1\leqslant 0$$

这个非线性规划问题，在没有约束条件下的解显然是$\mathbf{x}^*=(1, 1)^T$点，也就是：

$x_1=1$，$x_2=1$ 给出无约束极小值

$$F(1, 1)=0$$

而在加上三个不等式的约束要求以后，$\mathbf{x}=(x_1, x_2)^T$的可行域将是在第一象限单位圆所圈定的范围，即

$$x_1\geqslant 0, \quad x_2\geqslant 0$$

$$x_1^2+x_2^2\leqslant 1$$

要找出带约束的目标函数的极小值，先可以从目标函数各等值曲线

$$F(x_1, x_2)=C\Rightarrow(x_1-1)^2+(x_2-1)^2=C$$

的分析，在图 3-1 上可以看到它与单位圆相切的点$\mathbf{x}^*=$

$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^T$就是所求的带有约束的极小化目标函数的点。

事实上，由于目标函数和约束函数的对称性，这个例题的极小点应该落在

$$x_1=x_2, \quad g_3(x_1, x_2)=0\ \text{的第一象限位置}$$

所以把 $x_1^* = x_2^* = t$ 代入 $g_3(x_1, x_2) = 0$ 得到：

$$t > 0, \quad t^2 + t^2 - 1 = 0 \Rightarrow t = \frac{1}{\sqrt{2}}$$

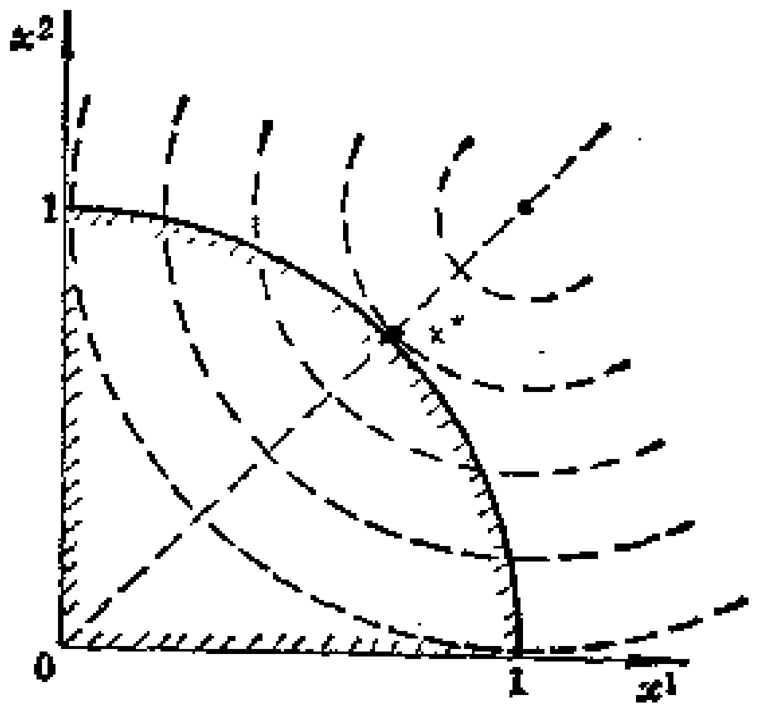


图3-1 带约束条件的极小点位置与可行域、等值曲线的关系

如果用 SUMT 的内点方法，写出例的惩罚函数，就 是 在保持取 $x_1 > 0$，$x_2 > 0$ 的情况下的函数：

$$P(\mathbf{x}, r^{(k)}) = F(\mathbf{x}) - \frac{r^{(k)}}{g_3(\mathbf{x})}$$

也就是：

$$P(\mathbf{x}, r^{(k)}) = (x_1 - 1)^2 + (x_2 - 1)^2 + \frac{r^{(k)}}{1 - x_1^2 - x_2^2}$$

为了说明 SUMT 的意义，设已知：

$$x_1 = x_2 = t$$

则惩罚函数可以写成一个带序列因子 $r^{(k)}$ 的一元函数：

$$F(t, r^{(k)}) = 2(t - 1)^2 + \frac{r^{(k)}}{1 - 2t^2}$$

在取 $r^{(k)} = 0$ 时，就是无约束极小化 $2(t-1)^2$ 一项的结果，即

$$\bar{x}_1^* = \bar{x}_2^* = \bar{t}^* = 1$$

在取 $r^{(k)} = 0.1$ 时，极小点由

$$F(t, 0.1) = 2(t - 1)^2 + \frac{0.1}{1 - 2t^2}$$

求得，将近似于 $x_1=x_2=0.6$ 点。

在取 $r^{(k)}=0.01$ 时，极小点由

$$F(t,\ 0.01)=2(t-1)^2+\frac{0.01}{1-2t^2}$$

求得，将近似于 $x_1=x_2=0.65$。

在取 $r^{(k)}$ 再趋小以后，极小点就将趋于

$$x_1^*=x_2^*=\frac{1}{\sqrt{2}}\approx 0.707$$

以上分析详见图 3-2。

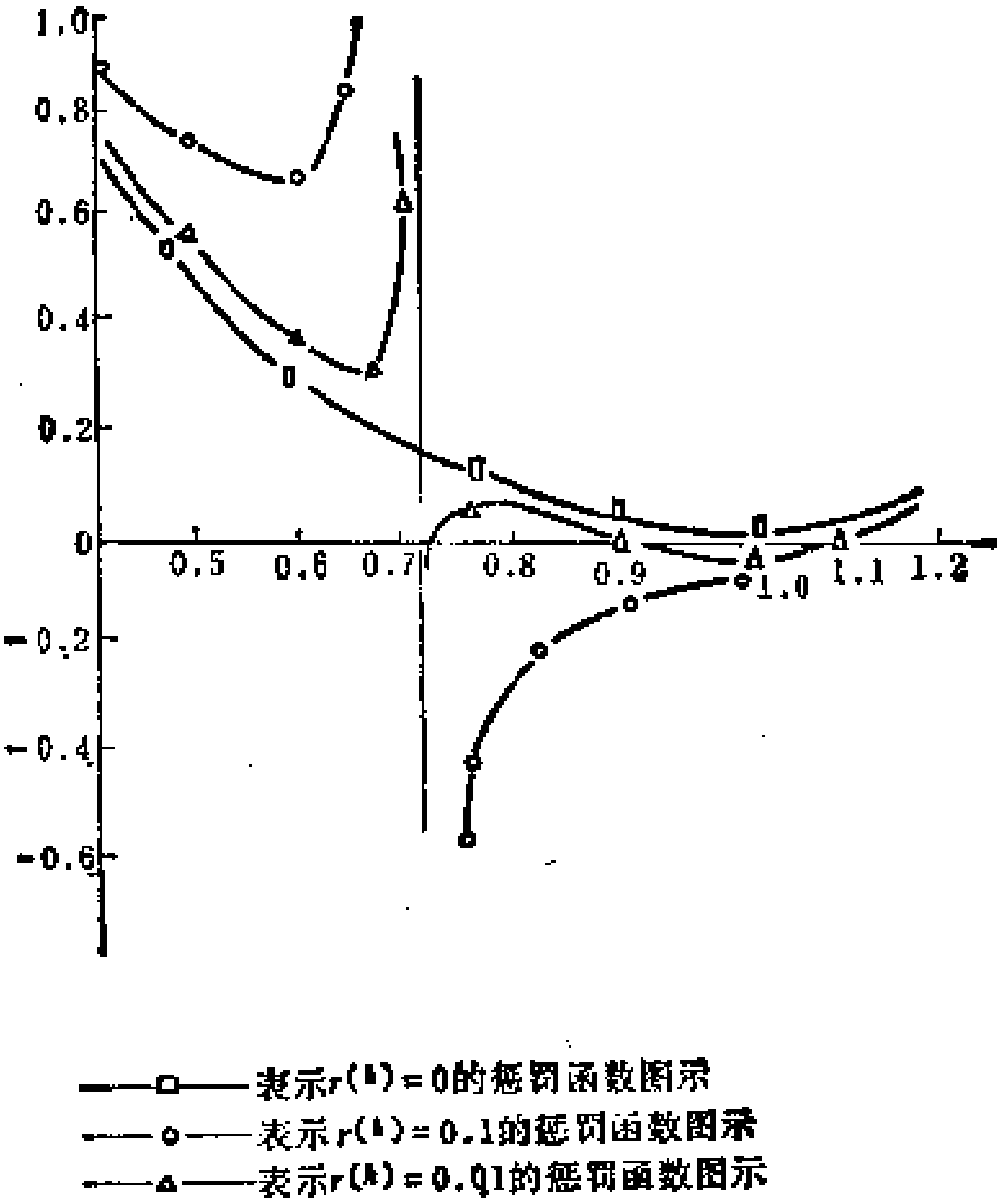
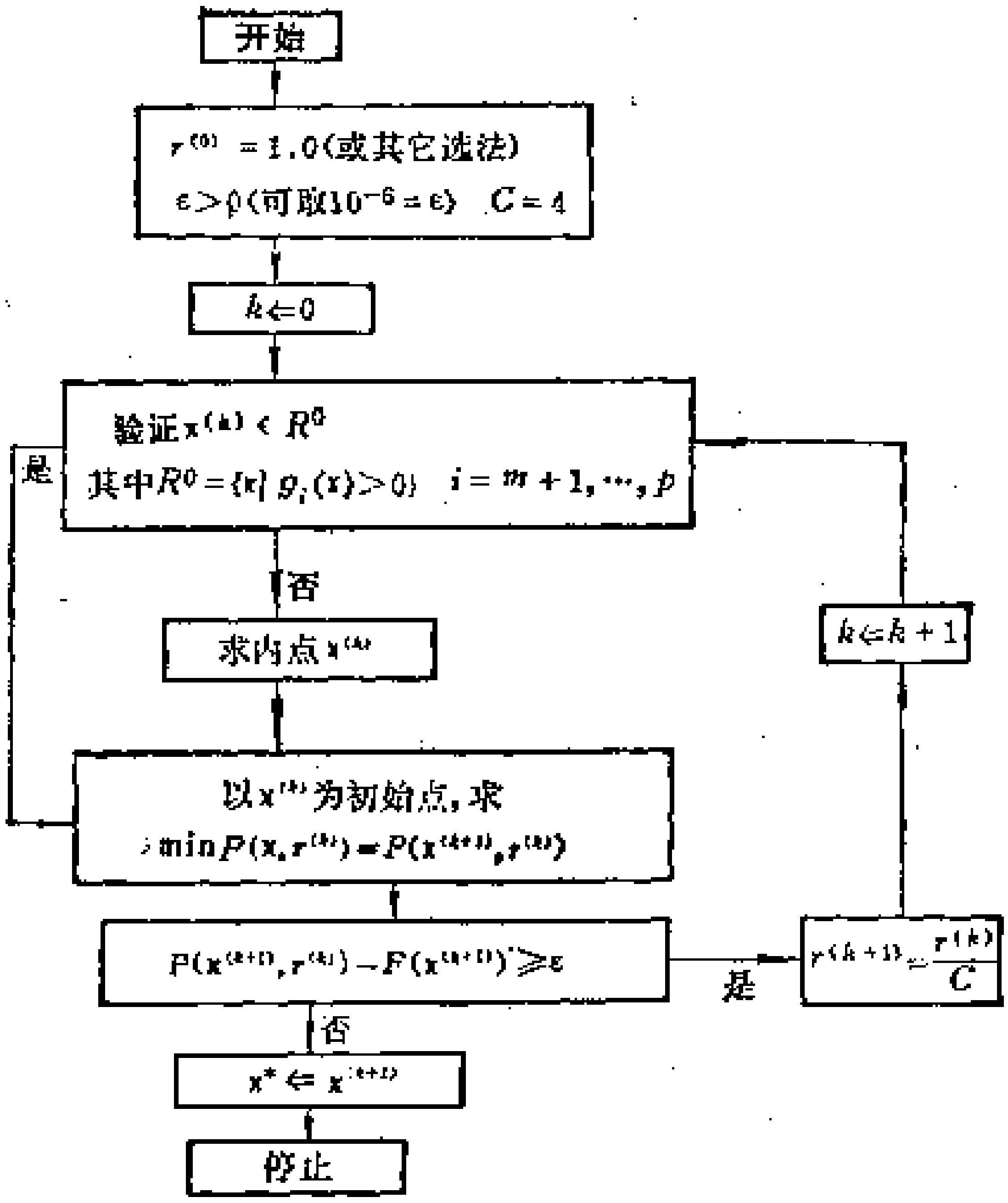


图3-2 例题的 SUMT 解法图示

总之，我们可以把 SUMT 的内点解法，列出框图如下：



或者，SUMT 内点算法用简略的语言写出：

（1）对 $k=0,\ 1,\ 2,\cdots$

1）$R^0=\{x|g_i(x)>0\quad i=m+1,\cdots,\ p\}$

2）若 $x^{(k)}\in R^0$，求

$$\min P(x,\ r^{(k)})=P(x^{(k+1)},\ r^{(k)})$$

3）$x^{(k)}\leftarrow x^{(k+1)}$

4）若 $x^{(k)}\notin R^0$，求 $\bar{x}^{(k)}\in R^0$

$x^{(k)}\leftarrow\bar{x}^{(k)}$ 转到 2）。

（2）对 $k=1,\ 2,\cdots$

1）$S(x^{(k)},\ r^{(k-1)})=P(x^{(k)},\ r^{(k-1)})-F(x^{(k)})$

2）若 $S<\varepsilon\ (\varepsilon>0)$ 停止，否则 $r^{(k)}\leftarrow r^{(k)}/C$ $(C>1)$。

从上所述可见，SUMT 内点解法的优点是计算上直观，几

乎每次计算都在可行域中进行。但是，在每次计算中，都要先验证下一步计算的起始点，或者说上一次求得的极小点是否是一个内点，否则内点法的障碍项就起不到障碍作用，从而不能使序列求出的极小点趋向某一定值。

然而，即使采用SUMT的内点算法，$r^{(k)}$很小以后，计出的序列极小点$\mathrm{x}^{(k)}$有时就是接近$g_i(\mathrm{x})=0$（$i=m+1,\cdots,p$）上的点，它的效果似乎和从外点（即非内点）接近的一样。所以，在一般的SUMT惩罚函数（3-5）中，我们可以采用外点消耗作用的项，即

$$G(g_i(\mathrm{x}))=\min(0,\ g_i(\mathrm{x}))^2$$

然后选逐渐趋大的序列正因子$r^{(k)}$形成SUMT的外点形式的惩罚函数，即

$$P(\mathrm{x},\ r^{(k)})=F(\mathrm{x})+r^{(k)}\left[\sum_{i=1}^{m}h_i^2(\mathrm{x})\right.$$

$$\left.+\sum_{i=m+1}^{p}\{\min(0,\ g_i(\mathrm{x}))\}^2\right] \qquad (3\text{-}8)$$

这样，我们把出现在外点的各个$g_i(\mathrm{x})$，和等式约束函数$h_i(\mathrm{x})$一样，使它们从平方量上消耗到接近于零。当然，使用SUMT外点算法（3-8），并不排斥计算过程中内点的出现，这时，我们只要消耗$h_i^2(\mathrm{x})$一组约束函数平方项就可以了。

下面，将SUMT外点算法用简略语言写出：

（1）对$k=0,\ 1,\ 2,\cdots$

1）求$\min P(\mathrm{x},\ r^{(k)})=P(\mathrm{x}^{(k+1)},\ r^{(k)})$

2）$\mathrm{x}^{(k)}\longleftarrow\mathrm{x}^{(k+1)}$

（2）对$k=1,\ 2,\cdots$

1）$S(\mathrm{x}^{(k)},\ r^{(k-1)})=P(\mathrm{x}^{(k)},\ r^{(k-1)})-F(\mathrm{x}^{(k)})$

2）若$S<\varepsilon$，停止，否则 $r^{(k)}\leftarrow Cr^{(k)}$（$C>1$）。

以上介绍的有关SUMT解法的内点算法（3-2）或（3-4）

及外点方法（3-8），它们的主要特点是在内点方法中，依靠障碍项

$$b(x)\equiv\sum_{i=m+1}^{p}\frac{1}{g_i(x)}\text{或}\quad b(x)\equiv-\sum_{i=m+1}^{p}\ln g_i(x)$$

限制所求解的点被围在可行域的附近；而在外点法中，尽量消耗不在可行域的现象就完全依赖于消耗项

$$l(x)\equiv\sum_{i=1}^{m}h_i^2(x)+\sum_{i=m+1}^{p}[\min(0,\ g_i(x))]^2$$

的作用。

如果按照序列无约束极小化技巧，即一般的 SUMT 的算法，几乎每一步都会得到所求极值点对一部分约束条件成立，而对另一部分并不成立。所以，最好的办法是对那些不成立的约束部分采用消耗项；而对成立的部分，特别是那些合于不等约束严格为不等的部分，采用障碍项。

这样，如果我们用 I_1 表示那些使不等式约束严格取不等关系的点所属的 i 附标的集合，用 I_2 表示其它的附标 i 的集合。写 SUMT 的惩罚函数是：

$$P(x,\ r^{(k)})=F(x)+r^{(k)}b(x)+(r^{(k)})^{-1}l(x) \qquad (3\text{-}9)$$

其中障碍项是：

$$b(x)=-\sum_{i\in I_1}\ln g_i(x)$$

而消耗项是：

$$l(x)=\sum_{i\in I_2}[\min(0,g_i(x))]^2+\sum_{j=1}^{m}h_j^2(x)$$

这种 SUMT 解法的序列因子 $r^{(k)}$ 仍是大于零的递减序列。称这种 SUMT 方法是一种混合法。

最后，将 SUMT 混合算法用简略语言写出：

（1）对 $k=0,\ 1,\ 2,\cdots$

1）若 $g_i(\mathbf{x}^{(k)})>0$， $i\in I_1$，否则 $i\in I_2$

2）求 $\min P(\mathbf{x},\ r^{(k)})=P(\mathbf{x}^{(k+1)},\ r^{(k)})$

3）$\mathbf{x}^{(k)}\leftarrow\mathbf{x}^{(k+1)}$

（2）对 $k=1,\ 2,\cdots$

1）$S(\mathbf{x}^{(k)},\ r^{(k-1)})=P(\mathbf{x}^{(k)},\ r^{(k-1)})-F(\mathbf{x}^{(k)})$

2）若 $S<\varepsilon$（$\varepsilon>0$）停止，否则 $r^{(k)}\leftarrow r^{(k)}/C$（$C>1$）。

关于 SUMT 解法的例子，除了在例 1 中我们已经举例说明了 SUMT 内点法各次求极小点的意义外，在第二章中，我们也就三杆构架的满约束设计给出了使用 SUMT 内点法的惩罚函数(2-27)。

但是，如何按每次给出的序列因子 $r^{(k)}$ 以求出相应的惩罚函数 $P(\mathbf{x},\ r^{(k)})$ 的极小点，即从 $\mathbf{x}^{(k)}$ 求出 $\mathbf{x}^{(k+1)}$ 的方法，又将要涉及到使用解非线性多元方程组的迭代解法，这就是我们要在第五章介绍的无约束最优解法的解析方法的问题。

§2 线性规划

数学规划问题式（3-1）中一种最简单的情况是目标函数为一个线性函数，约束条件统一写成线性不等式的形式，也就是线性规划问题，即

$$\text{极小化}\quad F(\mathbf{x})=c_1x_1+c_2x_2+\cdots+c_{n-m}x_{n-m}$$

使合于约束条件

$$\sum_{j=1}^{n-m}a_{ij}x_j\leqslant b_i\qquad i=1,\ 2,\cdots,\ m\tag{3-10}$$

$$x_i\geqslant 0\qquad i=1,\ 2,\cdots,\ n-m$$

这时，我们可以附加 m 个松弛变量 y_i（$i=1,\ 2,\cdots,\ m$），把式（3-10）中的不等式约束写成线性等式组：

$$\left.\begin{array}{l} a_{11}x_1+a_{12}x_2+\cdots+a_{1,n-m}x_{n-m}+y_1=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2,n-m}x_{n-m}+y_2=b_2 \\ \cdots\cdots \\ a_{m1}x_1+a_{m2}x_2+\cdots+a_{m,n-m}x_{n-m}+y_m=b_m \end{array}\right\} \quad (3\text{-}11)$$

其中

$$x_i \geqslant 0 \qquad i=1,\ 2,\cdots,\ n-m$$
$$y_i \geqslant 0 \qquad i=1,\ 2,\cdots,\ m$$

所以，经过松弛变量的加入，线性规划问题式（3-10）就可以写成如下的标准形式：

$$\begin{array}{ll} \text{极小化} & \mathbf{z}=\mathbf{c}^T\mathbf{x}=c_1x_1+c_2x_2+\cdots+c_nx_n \\ \text{使合于} & \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}\geqslant 0,\ \mathbf{b}\geqslant 0 \end{array} \quad (3\text{-}12)$$

这里的 $\mathbf{c}$ 是 n 向量或 $n\times 1$ 列矩阵：

$$\mathbf{c}=(c_1,\ c_2,\cdots,\ c_{n-m},\ c_{n-m+1},\cdots,\ c_n)^T$$

显然 $c_{n-m+1}=c_{n-m+2}=\cdots=c_n=0$；同时 $\mathbf{x}$ 是一个 n 向量或 $n\times 1$ 列矩阵：

$$\mathbf{x}=(x_1,\ x_2,\cdots,\ x_{n-m},\ x_{n-m+1},\cdots,\ x_n)^T$$

并且显然 $x_{n-m+1}=y_1, x_{n-m+2}=y_2,\cdots,x_n=y_m$。在写成 m 个等式约束中，$\mathbf{A}$ 表示一个 $m\times n$ 矩阵，也就是：

$$\mathbf{A}=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,n-m} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2,n-m} & 0 & 1 & \cdots & 0 \\ \cdots\cdots & & & & & & & \\ a_{m1} & a_{m2} & \cdots & a_{m,n-m} & 0 & 0 & \cdots & 1 \end{pmatrix}$$

而 $\mathbf{b}$ 是一个 m 向量或 $m\times 1$ 列矩阵：

$$\mathbf{b}=(b_1,\ b_2,\cdots,\ b_m)^T$$

在式（3-12）中，关于 $\mathbf{x}\geqslant o$，$\mathbf{b}>o$ 的意思就是 $\mathbf{x}$ 的各分量 $x_i\geqslant 0\ (i=1,\ 2,\cdots,\ n)$，$\mathbf{b}$ 的各分量 $b_i>0\ (i=1,2,\cdots,m)$。

欲使 b>o 成立，有时线性规划要如下提出：

$$极小化\quad F(x)=c_1x_1+c_2x_2+\cdots+c_{n-m}x_{n-m}$$

$$使合于\quad \sum_{j=1}^{n-m} a_{ij}x_j \geqslant b_i \qquad i=1,\ 2,\cdots,\ m \quad (3\text{-}13)$$

$$x_i \geqslant 0 \qquad i=1,\ 2,\cdots,\ n-m$$

则可以用剩余变量 $y_i(i=1,\ 2,\cdots,\ m)$使：

$$a_{i1}x_1+a_{i2}x_2+\cdots+a_{i,\,n-m}x_{n-m}-y_i=b_i$$

$$y_i \geqslant 0 \qquad i=1,\ 2,\cdots,\ m$$

从而又可以把式（3-13）写成式（3-12）的标准形式。

如果在 $x_i \geqslant 0$ 的要求中，某一个 x_i 是例外，例如 $x_1 \geqslant 0$ 不成立，也就是 x_1 可以任意取值时，可以令

$$x_1=u_1-v_1,\ u_1 \geqslant 0,\ v_1 \geqslant 0$$

或者把这种“自由变量”x_1从另外的变量关系中消去。详见下例。

例 2　极小化　$x_1+3x_2+4x_3$

使合于　$x_1+2x_2+x_3=5$

$2x_1+3x_2+x_3=6$

$x_2 \geqslant 0,\ x_3 \geqslant 0$

从两个等式约束得到：

$$x_1=5-2x_2-x_3 \Rightarrow x_2+x_3=4$$

所以问题转为：

$$极小化\quad x_2+3x_3$$

$$使合于\quad x_2+x_3=4$$

$$x_2 \geqslant 0,\ x_3 \geqslant 0$$

当然，对于这样简单的问题，我们可直接将

$$x_3=4-x_2$$

代入目标函数，使

$$F(x)=x_2+3x_3=12-2x_2=2(6-x_2)$$

极小，并使

$$x_3 = 4 - x_2 \geqslant 0$$

可见$x_2 = 4$，$x_3 = 0$，$x_1 = 5 - 2x_2 - x_3 = -3$将使原问题达极小值：

$$(-3) + 3 \times 4 + 4 \times 0 = 9$$

一般来说，式（3-12）的线性规划解是要最终找出它合于约束条件的目标函数 z 的极小点

$$\mathbf{x}^* = (x_1^*, \ x_2^*, \cdots, \ x_{n-m}^*, \ x_{n-m+1}^*, \cdots, \ x_n^*)^T$$

所给出的极小解，即

$$\mathbf{z}^* = \mathbf{c}^T\mathbf{x}^*$$

如果就式（3-10）或式（3-13）问题，直接使松弛量$y_i = b_i (i = 1, \ 2, \cdots, m)$或剩余变量$y_i = -b_i (i = 1, 2, \cdots, m)$，而令其它各变量为零，即

$$x_1 = x_2 = \cdots = x_{n-m} = 0$$

那么，它将是满足约束方程的，称为线性规划的“基本可行解”。基本可行解不一定可使目标函数达极小值，所以我们的目标，就是逐步的改变$n - m$个变量为零，找出另外m个变量不必为零的“基本解”，使线性规划问题得到最终的解。

如果我们把线性规划标准形式（3-12）的约束方程写成：

$$\begin{array}{llll}
x_1 & & + y_{1,m+1} x_{m+1} + \cdots + y_{1,n} x_n = y_{10} \\
& x_2 & + y_{2,m+1} x_{m+1} + \cdots + y_{2,n} x_n = y_{20} \\
& \ddots & \cdots \quad \cdots \\
& & x_m + y_{m,m+1} x_{m+1} + \cdots + y_{m,n} x_n = y_{m0}
\end{array}$$

并简列成表式，使首m列就是基本可行解

$$\mathbf{x}^T = (\mathbf{x}_B, \ 0) = (y_{10}, \ y_{20}, \cdots, y_{m0}, \ 0, \ 0, \cdots, \ 0)$$

对应的单位矩阵表出首n列，按列向量写出：

$$
\begin{array}{cccccccc}
\mathbf{a}^{(1)} & \mathbf{a}^{(2)} & \cdots & \mathbf{a}^{(m)} & \mathbf{a}^{(m+1)} & \cdots & \mathbf{a}^{(n)} & \mathbf{b} \\
1 & 0 & \cdots & 0 & y_{1,m+1} & \cdots & y_{1n} & y_{10} \\
0 & 1 & \cdots & 0 & y_{2,m+1} & \cdots & y_{2n} & y_{20} \\
\cdots & & \cdots & & & \cdots & & \\
0 & 0 & \cdots & 1 & y_{m,m+1} & \cdots & y_{mn} & y_{m0}
\end{array}
\qquad (3\text{-}14)
$$

在给定任一组 x_{m+1}, x_{m+2}, $\cdots$, x_n 数值以后，我们就可以按式(3-14) 写出：

$$
\begin{aligned}
x_1 &= y_{10} - \sum_{j=m+1}^{n} y_{1j}x_j \\
x_2 &= y_{20} - \sum_{j=m+1}^{n} y_{2j}x_j \\
&\cdots\cdots\cdots \\
x_m &= y_{m0} - \sum_{j=m+1}^{n} y_{mj}x_j
\end{aligned}
\qquad (3\text{-}15)
$$

再用式(3-15)，从目标函数

$$\mathbf{z} = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

内用 x_{m+1}, x_{m+2}, $\cdots$, x_n 写出 x_1, x_2, $\cdots$, x_m 就有：

$$
\begin{aligned}
\mathbf{z} = \mathbf{c}^T\mathbf{x} = z_0 &+ (c_{m+1} - z_{m+1})x_{m+1} + (c_{m+2} - z_{m+2})x_{m+2} \\
&+ \cdots + (c_n - z_n)x_n
\end{aligned}
\qquad (3\text{-}16)
$$

式中

$$z_0 = \mathbf{c}_B^T\mathbf{x}_B,\quad 其中：\mathbf{c}_B = (c_1,\ c_2, \cdots,\ c_m)^T$$

$$\mathbf{x}_B = (y_{10},\ y_{20}, \cdots,\ y_{m0})^T$$

$$z_j = y_{1j}c_1 + y_{2j}c_2 + \cdots + y_{mj}c_m \qquad m+1 \leqslant j \leqslant n$$

因此，只要基本可行解是一个非退化的解，即在 $\mathbf{x}_B$ 中没有任何 y_{i0}($i = 1,\ 2, \cdots,\ m$)为零时，可以从式(3-16)检查，如果对某 j 有

$$c_j - z_j < 0$$

的存在，我们就可以适当将 x_j($m+1 \leqslant j \leqslant n$)从原来是零的值

增加，而使基本解的某个变量从大于零的值改为零值，以 x_j 代替形成的新基本可行解，并使目标函数值减少。这种手续一直到所取的基本可行解构成的（3-16）式，所有的

$$c_j - z_j \geqslant 0 \qquad m+1 \leqslant j \leqslant n$$

就得到最优解。

在式（3-14）的表列中，各列系数构成向量 $\mathbf{a}^{(i)}$（$i=1, 2, \cdots, n$）和 $\mathbf{b}$，并且其中首 m 列相当于已经主元化计算以后得到的结果，也可以把 $\mathbf{a}^{(1)}$，$\mathbf{a}^{(2)}, \cdots$，$\mathbf{a}^{(m)}$ 看成是 m 维的正交正规基底向量，它们和式（2-8）所给出的 $\mathbf{e}^{(i)}$（$i=1, 2, \cdots, n$）是一样的，按式（2-14）内积的定义，它们都有这样的性质：

$$(\mathbf{a}^{(i)})^T(\mathbf{a}^{(j)}) = (\mathbf{a}^{(i)}, \mathbf{a}^{(j)}) = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$$

式中 $(\mathbf{a}^{(i)}, \mathbf{a}^{(i)}) = 1$ 表示

$$\|\mathbf{a}^{(i)}\| = (\mathbf{a}^{(i)}, \mathbf{a}^{(i)})^{1/2} = 1 \qquad i=1, 2, \cdots, m$$

是单位范数。

而 $(\mathbf{a}^{(i)}, \mathbf{a}^{(j)}) = 0$ （$i \neq j$）表示出一般意义下的正交性。

另外，我们可以用式（3-14）所示出的列向量将线性规划（3-12）的约束方程写成：

$$x_1\mathbf{a}^{(1)} + x_2\mathbf{a}^{(2)} + \cdots + x_m\mathbf{a}^{(m)} = \mathbf{b} \qquad (3-17)$$

并示出基本可行解。同时式（3-14）中任一列的系数向量将是正交正规基底向量 $\mathbf{a}^{(1)}$，$\mathbf{a}^{(2)}$，$\cdots$，$\mathbf{a}^{(m)}$ 的一个线性组合，即

$$\mathbf{a}^{(k)} = y_{1k}\mathbf{a}^{(1)} + y_{2k}\mathbf{a}^{(2)} + \cdots + y_{mk}\mathbf{a}^{(m)} \qquad (3-18)$$

$$m < k \leqslant n$$

将式（3-18）乘以 $\varepsilon \geqslant 0$，并用式（3-17）减之，就有：

$$(x_1 - \varepsilon y_{1k})\mathbf{a}^{(1)} + (x_2 - \varepsilon y_{2k})\mathbf{a}^{(2)} + \cdots + (x_m - \varepsilon y_{mk})\mathbf{a}^{(m)} + \varepsilon\mathbf{a}^{(k)} = \mathbf{b} \qquad (3-19)$$

$$m < k \leqslant n$$

所以，$\mathbf{b}$ 向量至多是 $m+1$ 个向量的线性组合，特别是，当

$\varepsilon=0$时，它就表出基本可行解（3-17）。

当ε从零增大时，$\mathbf{a}^{(k)}$的系数增加，在ε足够小的时候，(3-19)也可能给出一个可行解，但是它不是一个基本解。

这时，其它向量的系数有的随ε的增加而线性的减少，有的则随ε的增加而线性的增大，才能使式（3-19）成立，从而给出一组由$m+1$个变量的选定给出的可行解。如果在ε从零增加时，能有一个向量的系数减少，并且能有一个系数首先达到零时，也就是在取

$$\varepsilon=\min_{i}\{x_i/y_{ik},\ \ y_{ik}>0\} \tag{3-20}$$

时，就可用$\mathbf{a}^{(k)}$对应的变量取代$\mathbf{a}^{(i)}$中的某一个($i=1,\ 2,\cdots,\ m$)，从而对于已知单位矩阵化的$\mathbf{a}^{(1)}$，$\mathbf{a}^{(2)}$，…，$\mathbf{a}^{(m)}$系数向量而言，选

$$x_j/y_{jk}=y_{j0}/y_{jk} \tag{3-21}$$

的最小非负比做为取代变量的标准，在取代基本可行解的一个变量之后，就构成另一个基本可行解，然后再由式（3-16）判断是否需要进一步的改进。

由于我们所要求的解是可行解，所以：

$$x_i\geqslant 0\qquad i=1,\ 2,\ \cdots,\ m$$

从而，在基本可行解的计算求解过程中

$$x_i=y_{i0}\geqslant 0\qquad i=1,\ 2,\ \cdots,\ m$$

并且要使某$i=j$时

$$x_j-\varepsilon y_{jk}$$

能对$\varepsilon>0$趋向零，只有在$y_{jk}>0$的情况才有可能。所以在应用式（3-20）、（3-21）的判准法则时，如果没有y_{ik}正的存在，就没有什么新的基本可行解改善的可能。下面举例说明。

例 3　极大化　$z=3x_1+2x_2$

　　　使合于　$2x_1+x_2\leqslant 8$

　　　　　　　$x_1+3x_2\leqslant 15$

　　　　　　　$x_1\geqslant 0,\ x_2\geqslant 0$

也就是要求

$$\begin{aligned} &\text{极小化} \quad -z=-3x_1-2x_2 \\ &\text{使合于} \quad 2x_1+x_2\leqslant 8 \\ &\qquad\qquad x_1+3x_2\leqslant 15 \\ &\qquad\qquad x_1\geqslant 0, \quad x_2\geqslant 0 \end{aligned}$$

首先，按式（3-14）的形式列表，并把$-z$写在最后一行，它能表出c_j-z_j的结果。从而有：

	$a^{(1)}$	$a^{(2)}$	$a^{(3)}$	$a^{(4)}$	b
(1)	②	1	1	0	8
(2)	1	3	0	1	15
(3)	-3	-2	0	0	0

它表示出第一个基本可行解是：

$$x_1=0, \quad x_2=0, \quad x_3=8, \quad x_4=15$$

从式（3-16）则有：

$$-z=z_0+(c_1-z_1)x_1+(c_2-z_2)x_2$$

$$z_0=c_3\cdot 8+c_4\cdot 15=0\times 8+0\times 15=0$$

$$z_1=2\cdot c_3+1\cdot c_4=2\times 0+1\times 0=0$$

$$z_2=1\cdot c_3+3\cdot c_4=1\times 0+3\times 0=0$$

所以 $\quad -z=0+(-3)x_1+(-2)x_2$

也就是 $\quad c_1-z_1<0, \quad c_2-z_2<0$

从而有改进的可能。也就是再从现在的$y_{10}=8$，$y_{20}=15$可得，

$$\frac{y_{10}}{y_{11}}=\frac{8}{2}=4 \quad \text{（在 } y_{11} \text{ 加上圈号，备查）}$$

此值是非负的最小值，从（3-20）得知，可用x_1代x_3，将（1）除以2得（4），（2）减（4）得（5），（3）加上（4）乘3得出（6），于是得到：

	$a^{(1)}$	$a^{(2)}$	$a^{(3)}$	$a^{(4)}$	b
(4)	1	1/2	1/2	0	4
(5)	0	5/2（圈）	$-1/2$	1	11
(6)	0	$-1/2$	3/2	0	12

给出基本可行解是：

$$x_1=4,\ x_2=0,\ x_3=0,\ x_4=11$$

它使

$$-z=z_0+(c_2-z_2)x_2+(c_3-z_3)x_3$$

$$z_0=c_1\cdot 4+c_4\cdot 11=-3\times 4=-12$$

$$c_2-z_2=-2-\left[\frac{1}{2}(-3)+\frac{5}{2}\times 0\right]=-\frac{1}{2}<0$$

并且

$$\frac{y_{20}}{y_{22}}=\frac{11}{\frac{5}{2}}=\frac{22}{5}$$

是正的最小，故应以 x_2 代 x_4，正规化后得：

	$\mathbf{a}^{(1)}$	$\mathbf{a}^{(2)}$	$\mathbf{a}^{(3)}$	$\mathbf{a}^{(4)}$	$\mathbf{b}$
(7)	1	0	3/5	−1/5	9/5
(8)	0	1	−1/5	2/5	22/5
(9)	0	0	7/5	1/5	11/5

给出基本可行解是：

$$x_1=9/5,\ x_2=22/5,\ x_3=0,\ x_4=0$$

并且

$$-z=z_0+(c_3-z_3)x_3+(c_4-z_4)x_4$$

其中 $z_3=3/5(-3)+(-1/5)(-2)<0,\ c_3-z_3>0$

$z_4=-1/5(-3)+2/5(-2)<0,\ c_4-z_4>0$

所以最优基本可行解是：

$$-z=z_0=c_1\cdot 9/5+c_2\cdot 22/5$$

$$=(-3)\cdot\frac{9}{5}+(-2)\cdot\frac{22}{5}=-71/5$$

也就是问题的解是（见图 3-3）：

$$x_1=\frac{9}{5},\quad x_2=\frac{22}{5},\quad z=\frac{71}{5}$$

例 3　这种解线性规划的方法常被称为单形方法，在图 3-3

中所示的阴影部分表示问题的可行域，而目标函数的等值线 $z=0$，$z=12$，$z=\frac{71}{5}$ 各示出了基本可行解所经过的点(0，0)，(4，0)，$\left(\frac{9}{5},\ \frac{22}{5}\right)$。

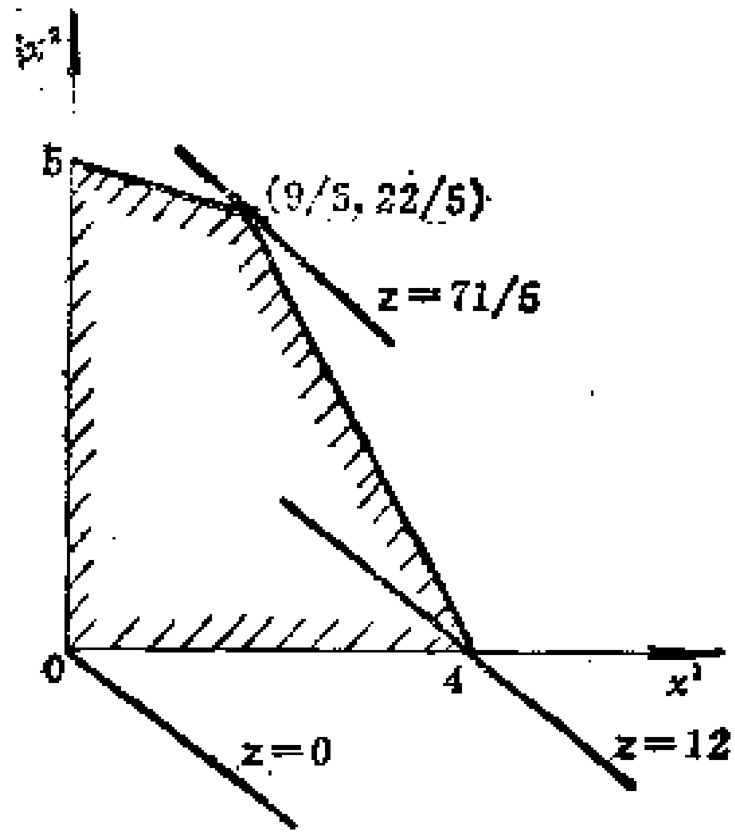


图3-3　例3的求解过程

“线性规划方法”是解决数学规划问题中最好的一种方法，它和通常的“优选法”是“最优化的数学方法”的两个重要方法，是早在五十年代就形成的。

§8　近似线性规划

由于线性规划解法比较完善，也能应用到比较多维的情况，所以就有必要考虑使用线性规划解法的技巧来近似地解决非线性规划问题。

设：给出一个非线性规划问题，

$$\left.\begin{array}{lll}\text{极小化} & F(x),\ x\in D\subset R^n & \\ \text{使合于} & h_i(x)=0 & i=1,\ 2,\ \cdots,\ m \\ & g_i(x)\geqslant 0 & i=m+1,\ \cdots,\ p\end{array}\right\}\quad(3\text{-}22)$$

如果 $x^{(k)}$ 已知是接近于非线性规划解 x^* 的点，我们可以把非

线性目标函数$F(\mathbf{x})$，约束函数$h_i(\mathbf{x})$及$g_i(\mathbf{x})$各在$\mathbf{x}^{(k)}$点做泰勒展式的一次近似，给出小步序的梯度方法，将问题在$\mathbf{x}^{(k)}$点线性化为：

$$\left.\begin{aligned}
&\text{极小化}\quad F(\mathbf{x})=F(\mathbf{x}^{(k)})+\sum_{j=1}^{n}\frac{\partial F(\mathbf{x}^{(k)})}{\partial x_j}\Delta x_j^{(k)}\\
&\qquad\qquad \Delta x_j^{(k)}=x-x_j^{(k)}\\
&\text{使合于等式约束}\quad \sum_{j=1}^{n}\frac{\partial h_i(\mathbf{x}^{(k)})}{\partial x_j}\Delta x_j^{(k)}=-h_i(\mathbf{x}^{(k)})\\
&\qquad\qquad i=1,\ 2,\ \cdots,\ m\\
&\text{不等式约束}\quad \sum_{j=1}^{n}\frac{\partial g_i(\mathbf{x}^{(k)})}{\partial x_j}\Delta x_j^{(k)}\geqslant -g_i(x^{(k)})\\
&\qquad\qquad i=m+1,\ \cdots,\ p
\end{aligned}\right\}\quad(3\text{-}23)$$

并且使线性化（3-23）的解$\mathbf{x}^{(k+1)}$再作（3-23）在$\mathbf{x}^{(k+1)}$点的展开近似求解，使相继问题的解的极限趋于非线性规划问题（3-22）的解。

例 4

$$\left.\begin{aligned}
&\text{极小化}\quad F(\mathbf{x})=x_1^2+x_2^2-16x_1-10x_2\\
&\text{使合于}\quad g_1(\mathbf{x})=11-x_1^2+6x_1-4x_2\geqslant 0\\
&\qquad\qquad g_2(\mathbf{x})=x_1x_2-3x_2-e^{x_1-3}+1\geqslant 0\\
&\qquad\qquad g_3(\mathbf{x})=x_1\geqslant 0\\
&\qquad\qquad g_4(\mathbf{x})=x_2\geqslant 0
\end{aligned}\right\}\quad(3\text{-}24)$$

图 3-4 的虚线表示出$F(\mathbf{x})$等值线，曲实线表示非线性约束$g_1(\mathbf{x})=0$，$g_2(\mathbf{x})=0$，轴线表示$g_3(\mathbf{x})=0$，$g_4(\mathbf{x})=0$，虚直线表示近似的目标函数等值线。

它的真解是在极值点

$$\mathbf{x}^*=(5.27,\ 3.68)^T$$

上的极小化目标函数$F(\mathbf{x}^*)=-79.9$。

设$\mathbf{x}^{(0)}=(4,3)^T$是初始可行点，它的目标函数值是$F(\mathbf{x}^{(0)})=-69$，可以写出它的第一次线性近似规划，即

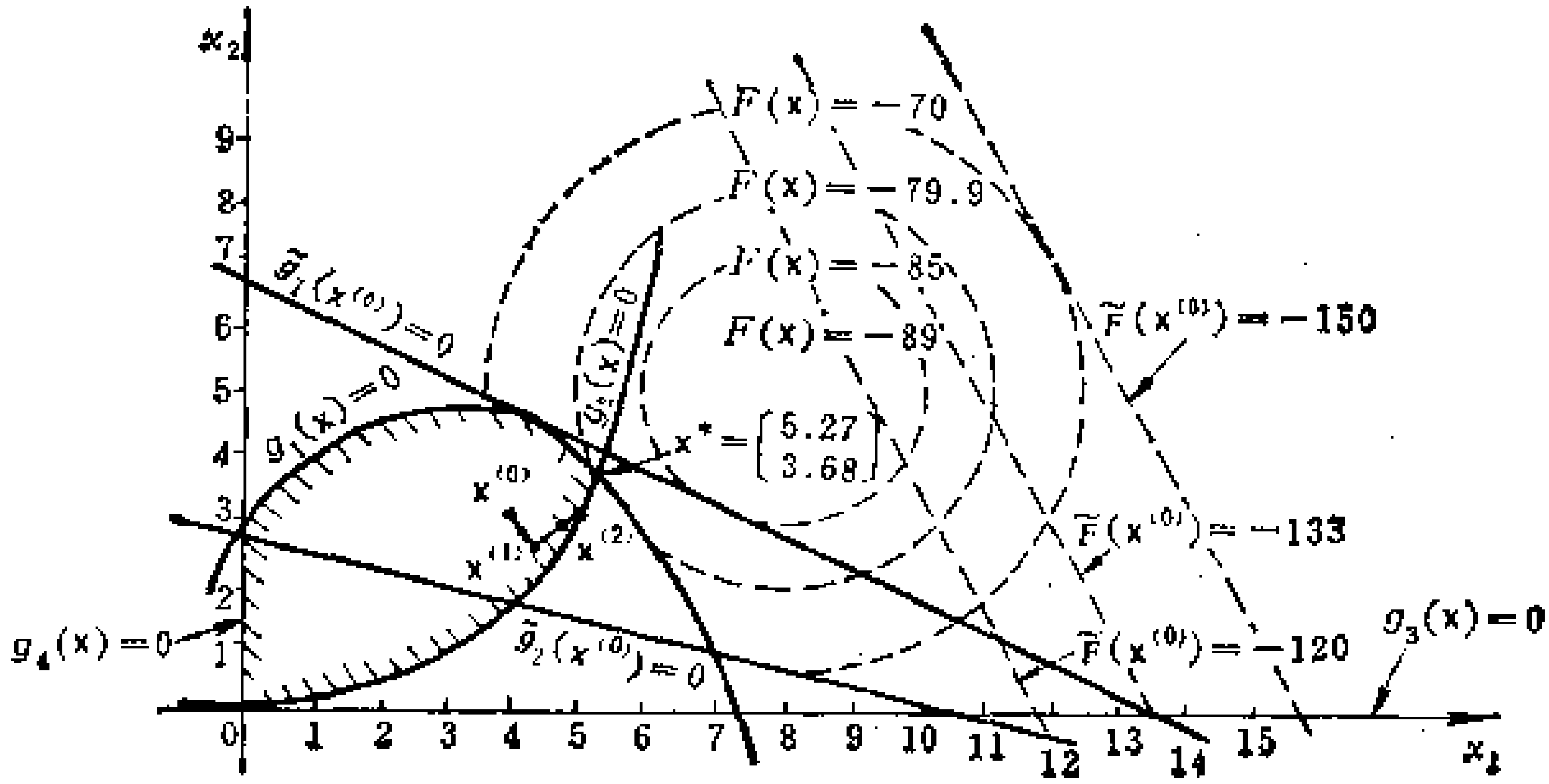


图3-4　例4近似线性解图示

$$
\left.
\begin{aligned}
\text{极小化}\quad & \tilde{F}(x^{(0)})=-8x_1-4x_2-25 \\
\text{使合于}\quad & \tilde{g}_1(x^{(0)})=-2x_1-4x_2+27\geqslant 0 \\
& \tilde{g}_2(x^{(0)})=0.28x_1+x_2-2.84\geqslant 0 \\
& \tilde{g}_3(x^{(0)})=x_1\geqslant 0 \\
& \tilde{g}_4(x^{(0)})=x_2\geqslant 0
\end{aligned}
\right\}
\qquad (3\text{-}25)
$$

它的解是 $x^{(1)}=(13.5,\ 0)^T$。虽然它对于式（3-25）来说是可行解，但是对于式（3-24）原问题来说却并不是一个可行域内的解。

为了防止x的解的近似值从（3-24）的问题的可行域离开，必须加以限制，即近似求出的解 $x^{(1)}$ 应合于一定条件，例如：

$$|x_j^{(1)}-x_j^{(0)}|\leqslant\delta_j^{(0)}\qquad j=1,\ 2,\ \cdots,\ n\qquad (3\text{-}26)$$

再从问题（3-24）的边界条件的限制，可以看出适宜选取：

$$\delta^{(0)}=[0.5,\ 0.5]^T$$

因此，从 $x^{(0)}=(4,3)^T$ 到 $x^{(1)}=(13.5,\ 0)^T$ 是使 x_1 增加，x_2 减少。所以由（3-26）的限制，加以近似求（3-25）解的结果，可以进一步修改为：

$$x^{(1)}=(4+0.5,\ 3-0.5)^T=(4.5,\ 2.5)^T$$

所得目标函数值是 $F(\mathbf{x}^{(1)})=-70.65$。

第二步，从 $k=1$ 开始，并且更缩小条件限制范围，使

$$\delta^{(1)}=0.8\delta^{(0)}=(0.4,\ 0.4)^T$$

并将式（3-24）在 $\mathbf{x}^{(1)}=(4.5,\ 2.5)^T$ 近似线性化为：

$$\begin{aligned} &\text{极小化} \quad \tilde{F}(\mathbf{x}^{(1)})=-7x_1-5x_2-36 \\ &\text{使合于} \quad \tilde{g}_1(\mathbf{x}^{(1)})=31.4-3x_1-4x_2\geqslant 0 \\ &\qquad\qquad \tilde{g}_2(\mathbf{x}^{(1)})=5.42-1.98x_1+1.5x_2\geqslant 0 \qquad (3\text{-}27) \\ &\qquad\qquad \tilde{g}_3(\mathbf{x}^{(1)})=x_1\geqslant 0 \\ &\qquad\qquad \tilde{g}_4(\mathbf{x}^{(1)})=x_2\geqslant 0 \end{aligned}$$

则式（3-27）的基本可行解是 $\mathbf{x}^{(2)}=(5.45,\ 3.58)^T$，它是很接近于式（3-24）的解的。再加条件

$$|x_j^{(2)}-x_j^{(1)}|\leqslant\delta_j^{(1)} \qquad j=1,\ 2,\ \cdots,\ n$$

限制，现在 $n=2$，应该得近似解为：

$$\begin{aligned} \mathbf{x}^{(2)}&=(4.5+0.4,\ 2.5+0.4)^T \\ &=(4.9,\ 2.9)^T \end{aligned}$$

从而目标函数极小值的近似值是：

$$F(\mathbf{x}^{(2)})=-75.1$$

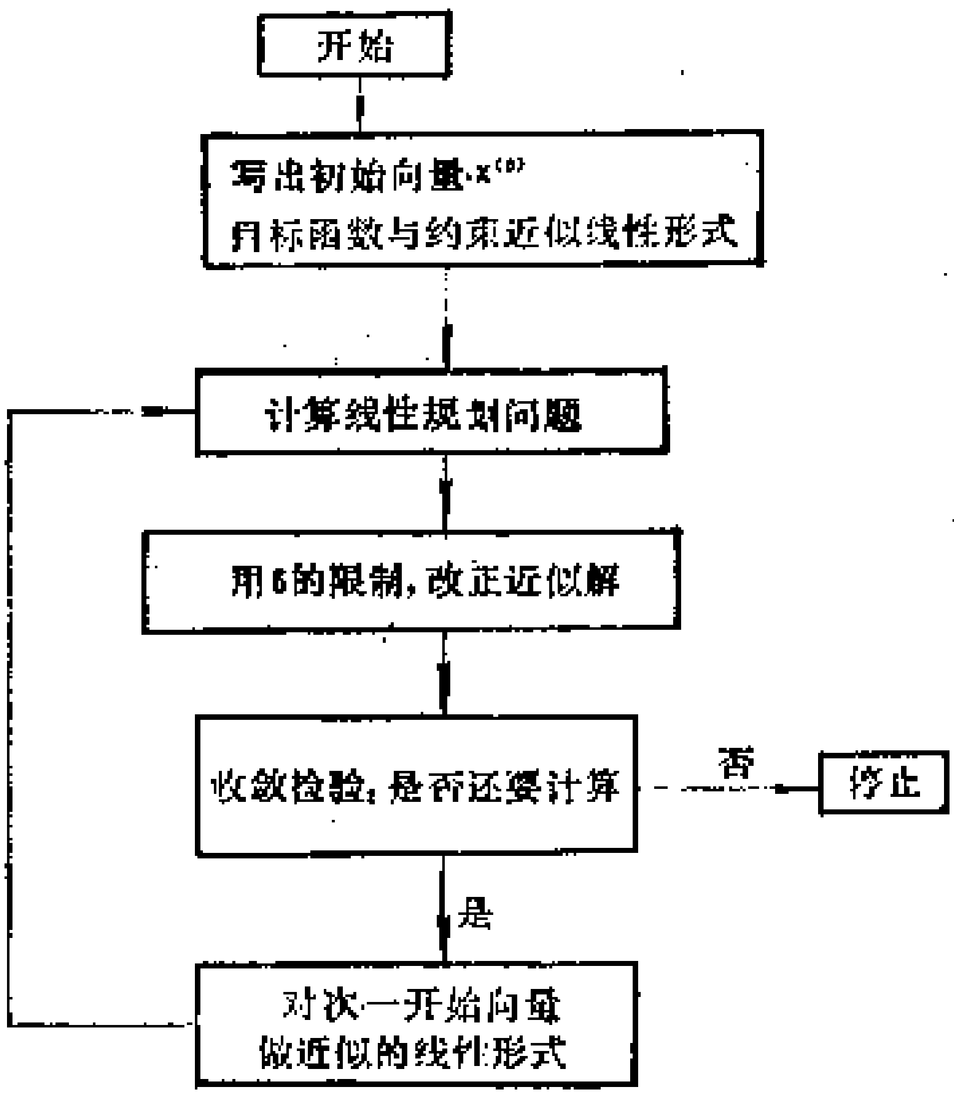
如此继续下去，直到所得 $\mathbf{x}^{(k)}$ 的解稳定，即合于

$$|x_j^{(k+1)}-x_j^{(k)}|<\varepsilon$$

$$j=1,\ 2,\ \cdots,\ n$$

为止。

所以，利用各初试点进行的近似线性规划解的过程，可概括如右框图。



另一种利用线性规划求解的近似方法，是

把变量按可分离形式，将目标函数写成：

$$F(\mathbf{x})=F_1(x_1)+F_2(x_2)+\cdots+F_n(x_n) \qquad (3\text{-}28)$$

然后，把每一个 $F_j(x_j)$ 这种一元函数用点插值方法，分别表示成线性形式，相应的改变约束条件，构成一种“可分离的规划”。

例5　极大化　$F(x)=2x_1-x_1^2+10x_2-2x_2^2$

关于第二个函数，它是定义在（0，4）内，将x_2分成两段，有：

$$x_2=v_1+v_2$$
$$0\leqslant v_1\leqslant 2$$
$$0\leqslant v_2\leqslant 2\ (=4-2)$$

并在（0，2）中用oC线段函数表示，在（2，4）中用CD线段函数表示，如图3-5(b)所示那样，则第二个函数的近似函数是：

$$\widetilde{F}_2=6v_1-2v_2$$

所以，总的目标函数可改变为近似的线性函数与相应的线性约束的线性规划问题，即

极大化 $\widetilde{F}=5u_1+0\cdot u_2+6v_1-2v_2$

使合于 $u_1+u_2+v_1+v_2\leqslant 5$

$u_1+u_2+2v_1+2v_2\leqslant 8$

$u_1\leqslant 3,\quad u_2\leqslant 2,\quad v_1\leqslant 2,\quad v_2\leqslant 2$

$u_1\geqslant 0,\quad u_2\geqslant 0,\quad v_1\geqslant 0,\quad v_2\geqslant 0$

用线性规划的单形法可解得：

$$u_1=3,\quad u_2=0,\quad v_1=2,\quad v_2=0,\quad \widetilde{F}=27$$

也就是原给出的非线性规划的近似解是：

$$x_1=3,\quad x_2=2,\quad \widetilde{F}=27$$

而且它实际上正是原问题的确切解。

在有些线性规划问题中，关于变量或其中某些变量，要求必须取非负整数（0，1，2，…）的解，这种问题称为“整线性规划”或“离散线性规划”。它和线性规划的区别只是在约束条件中再加以整值点的要求。

例6　极大化 $z=3x_1+2x_2$

使合于 $2x_1+3x_2\leqslant 14$

$2x_1+x_2\leqslant 9$

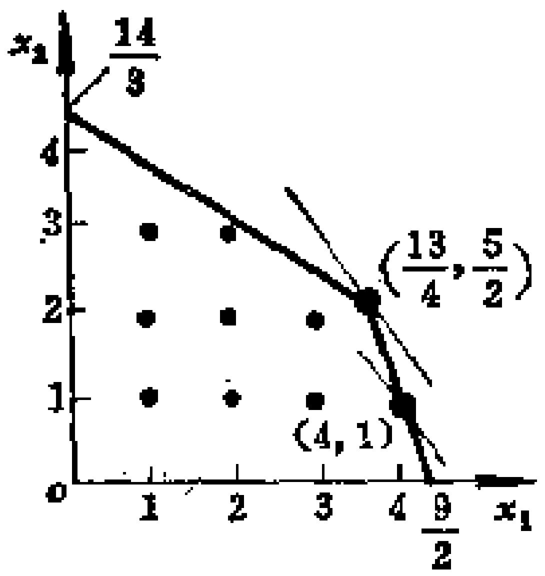
$x_1\geqslant 0,\ x_2\geqslant 0$ 的整值

首先，用线性规划的单形法求解，并且在列表方式上也可以改变形式，用 x_0，x_1，x_2，x_3，x_4 各表示原来 b，$a^{(1)}$，$a^{(2)}$，$a^{(3)}$，$a^{(4)}$所对应的系数列，第一行表示出原来待极大化的函数系数，第五行列出待极小化的对应函数，表头表示基本变量与对应的目标函数中的系数（例如 I 中，x_3，x_4 为基本变量，它们在目标函数中的系数各为 0，0），则单形方法的计算过程可表示为：

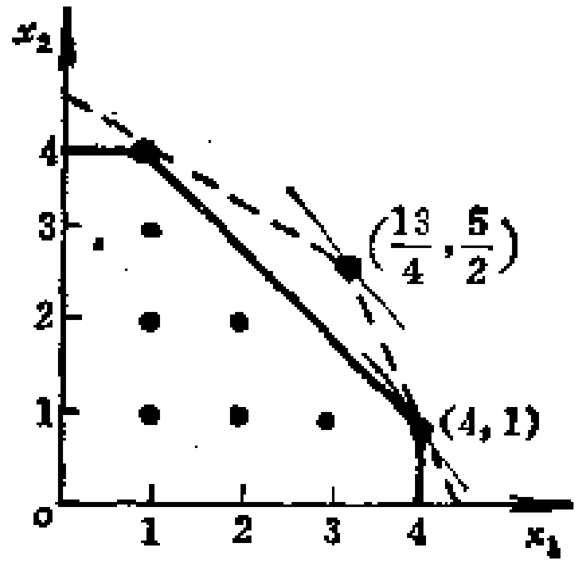
			0	3	2	0	0
			x_0	x_1	x_2	x_3	x_4
I	0	x_3	14	2	3	1	0
	0	x_4	9	2	1	0	1
	c_j-z_j		0	−3	−2	0	0
II	0	x_3	5	0	2	1	−1
	3	x_1	9/2	1	1/2	0	1/2
	c_j-z_j		27/2	0	−1/2	0	3/2
III	2	x_2	5/2	0	1	1/2	−1/2
	3	x_1	13/4	1	0	−1/4	3/4
	c_j-z_j		59/4	0	0	1/4	1/4

所以最优非负解是：

$$x_1=13/4,\quad x_2=5/2,\quad x_3=0,\quad x_4=0,\quad z=59/4$$



(a)



(b)

图3-6　整线性规划最优解

由图 3-6（a）所示，从中相近地选取（只对 x_1，x_2 选取）为：

$$x_1=4,\ x_2=1,\ x_3=0,\ x_4=0,\ z=14$$

实际上，它相当于在原来可行域中，切割掉最边缘部分的非整数部分，用整数点圈成边界，即不等式线性约束改为：

$$x_1+x_2\leqslant 5$$
$$x_1\leqslant 4$$
$$x_2\leqslant 4$$
$$x_1\geqslant 0$$
$$x_2\geqslant 0$$

也就是图 3-6（b）所示的情况，它表出了整线性规划的凸可行域，从而式（4，1）是它的一个角点，最优解自然是通过这个角点的等值线所给出的目标函数值。

§4 动态规划

上述各节，可以看出数学规划的特点，主要的是同时对所有变量在一定条件下求解。但是，实际构成问题的本身常常是分阶段的。

例如：线性规划问题，

$$\text{极大化}\quad z=8x_1+10x_2$$
$$\text{使合于}\quad 4x_1+2x_2\leqslant 12$$
$$x_1\geqslant 0,\ x_2\geqslant 0$$

可以是解决最优容量利用的问题，x_1 和 x_2 是由两个部分每个周期生产的数量，每个部分各生产一个单位需要 4 个机器小时和 2 个机器小时，边界约束条件表示每周期的最大容许的机器时刻，而目标函数的系数表示两个部分生产一个单位的收益。

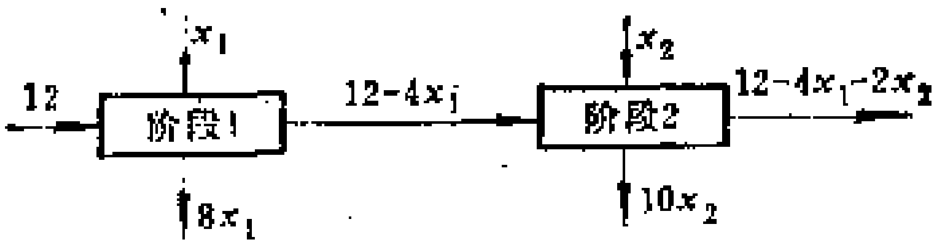
用线性规划的单形法可解得：

$$x_1=0,\ x_2=6,\ z=60$$

现在可以把问题分解为一种序列阶段，每一阶段仅对应于一个变量待以解决的“子问题”。从各个子问题以决定整个序列在

各个时刻的各个变量值。

若将第一个部分表为阶段 1，12 个容量因素（机器小时）是限制条件，如果生产了 x_1 单位，则留下了 $(12-4x_1)$ 个机器小时给第二个阶段，如下表示出：



在线性规划中所谓松弛变量就是表示不能利用的容量，而 $8x_1$ 和 $10x_2$ 各表示整个收益。

我们可以采用从后推向前的方式来求解这种问题，把 x_1 当做一个参数，暂时不动，将第 2 阶段先对 x_2 最优化。在取参数 $x_1=\bar{x}_1$ 时，使阶段 2 的最优容量从约束条件可以看出是 $12-4\bar{x}_1$ 机器小时，所以带参数的阶段 2 的子问题是：

$$\begin{aligned}&\text{极大化}\quad z_1=10x_2\\&\text{使合于}\quad 2x_2\leqslant 12-4\bar{x}_1\\&\qquad\qquad\ x_2\geqslant 0\end{aligned}$$

它也是一个线性规划问题，它的解显然是：

$$x_2=6-2\bar{x}_1,\quad z_{1,\max}=60-20\bar{x}_1$$

其次，对它的判定变量 x_1 来最优化阶段 1，可利用的容量是 12 机器小时，x_1 单位的生产能给出 $8x_1$ 收益，同时第 2 阶段的整个收益 $z_{1,\max}=60-20x_1$ 也依赖于 x_1，所以要求：

$$\begin{aligned}&\text{极大化}\quad z_2=8x_1+z_{1,\max}=60-12x_1\\&\text{使合于}\quad 4x_1\leqslant 12\\&\qquad\qquad\ x_1\geqslant 0\end{aligned}$$

它也是一个线性规划问题，由于 z_2 是 x_1 的一个减函数，所以直接得解：

$$x_1=0,\quad z_{2,\max}=60$$

从而给出结果：

$$x_2=6-2x_1=6$$

这种把一个问题分阶段的分成序列问题来求解的方法，就是“动态规划”的一个例子。

一般的动态规划并不限于原问题是线性规划，更不限于子问题是与原问题是同类型的数学规划。

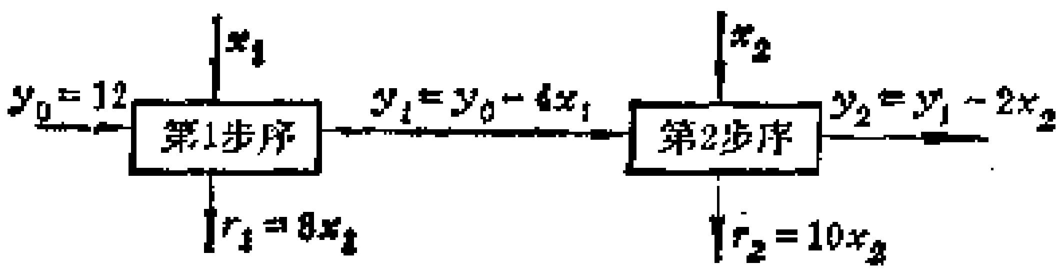
在我们把一个最优化问题化为动态规划求解的多阶段问题时，可以就各个阶段标以 $n=1,2,\cdots,N$，连带的引出一种状态变量 y_n。那么上例提出的生产过程，可以看成从现有的 $y_0=12$ 机器小时的一种给定初态开始，第一部分生产 x_1 个单位，每单位需要 4 个机器小时，所以阶段 1 的输出状态是：

$$y_1=y_0-4x_1=12-4x_1$$

它也是阶段 2 的输入状态。经过第二部分生产 x_2 个单位，还可以利用的容量是：

$$y_2=y_1-2x_2=y_0-4x_1-2x_2$$

它表出了最末的状态，从定义可知 $y_2\geqslant 0$。也就是如下的过程：



一般的情况，第 n 个步序的输入状态 y_{n-1} 变换到一个输出状态 y_n，它的变化是由判定变量 x_n 来带动的，所以整个系统的状态变化可写成：

$$y_n=f_n(y_{n-1},\ x_n)\qquad n=1,\ 2,\cdots,n \tag{3-29}$$

上例中的 $y_1=y_0-4x_1$，$y_2=y_1-2x_2$，一般的情况是某一种 f_n 的函数形式。式（3-29）称为“变换方程组”。

每一步序的收益，就是各个步序对目标函数 z 的贡献。上例中的 $r_1=8x_1$ 和 $r_2=10x_2$ 就是表示上例两个步序中各个收益。一般的情况，依赖于输入状态和判定变量的这个“代价”函数是：

$$r_n=r_n(y_{n-1},\ x_n)\qquad n=1,\ 2,\cdots,N \tag{2-30}$$

计算的第一步是从生产系统的最后步序开始，由后向前递推的 N 个步序是：

$$\left.\begin{aligned}
&z_1=r_N(y_{N-1},\ x_N)\\
&R_1(y_N)=\max_{x_n}[r_N(y_{N-1},\ x_N)]\\
&z_2=r_{N-1}(y_{N-2},\ x_{N-1})+R_1(y_{N-1})\\
&y_{N-1}=t_{N-1}(y_{N-2},\ x_{N-1})\\
&\cdots\cdots\\
&z_N=r_1(y_0,\ x_1)+R_{N-1}(y_1)\\
&y_1=t_1(y_0,\ x_1)
\end{aligned}\right\}\qquad(3\text{-}31)$$

并且 x_n 的决定依赖于 y_{N-1}，也就是最终将得到参数状态解，即

$$x_n=x_n(y_{n-1})\qquad n=N,\ N-1,\cdots,1\qquad(3\text{-}32)$$

它的计算过程如下：

$$\begin{tikzcd}
y_0 \arrow[rr] \arrow[dr] & & y_1=t_1(y_0,x_1) \arrow[rr] \arrow[dr] & & y_2=t_2(y_1,x_2) \arrow[rr] \arrow[dr] & & \cdots \\
& x_1(y_0) \arrow[ur] & & x_2(y_1) \arrow[ur] & & x_3(y_2) \arrow[ur] &
\end{tikzcd}$$

还可以用相应的从前向后递推计算的动态规划方案，这时，由

$$y_n=t_n(y_{n-1},\ x_n)$$

得

$$y_{n-1}=t_n^*(y_n,\ x_n)$$

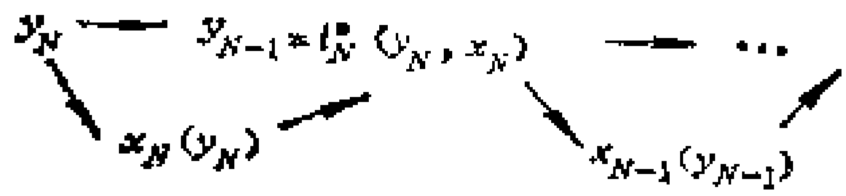
所以有：

$$\begin{aligned}
&z_1=r_1(y_0,\ x_1)\\
&R_1(y_1)=\max_{x_1}[r_1(y_0,\ x_1)]=\max_{x_1}[r_1^*(y_1,\ x_1)]\\
&y_0=t_1^*(y_1,\ x_1)\qquad\qquad(3\text{-}33)\\
&z_2=r_2(y_1,\ x_2)+R_1(y_1)\\
&R_2(y_2)=\max_{x_2}[r_2^*(y_2,\ x_2)]\\
&y_1=t_2^*(y_2,\ x_2)\\
&\cdots\cdots\\
&z_N=r_N(y_{N-1},\ x_N)+R_{N-1}(y_{N-1})\\
&y_{N-1}=t_N^*(y_N,\ x_N)
\end{aligned}$$

最后得到参数状态解是：

$$x_n = x_n(y_n) \qquad n = 1, \ 2, \ \cdots, \ N \qquad (3\text{-}34)$$

其过程如下：



现在，我们用最简单的"最短路径"的问题来说明动态规划的一种直接应用。

例如：若一个路径经过图形式化以后的形状如图 3-7 所示，驾驶员怎样才能找到从P到Q的最短路径呢？

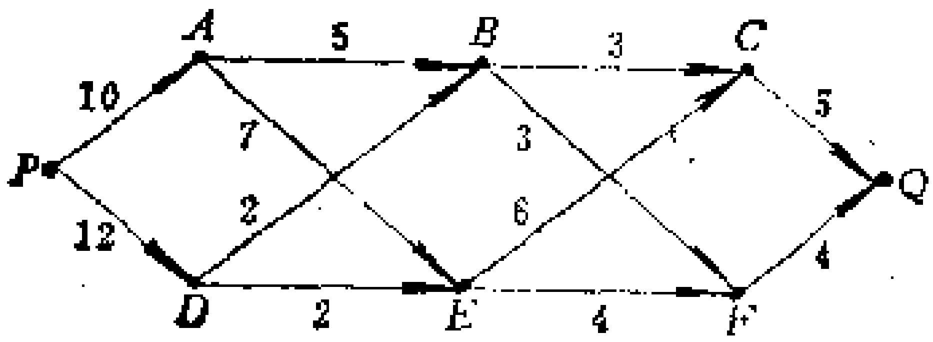


图3-7 从P到Q的路径图

现在从P到Q之间有六个中间站A，B，C，D，E，F，它们彼此之间存在的通路和路径之长都标在图中，而且设定行驶的方向总是从左到右，譬如从B不能再回到A或D，但是可以到C或F。

任一个从P到Q的路径要经过三次相继的判定。从P开始，驾驶员必须决定是到A或D，如果他选定是行驶到A，那么从A就要决定是到B或E，而每一个结果又要有到C或F的决定，到了C或到了F之后，就不再有什么选择的到达终点Q。

由于每一种判定是在两种不同情况之间的一种选择，而且有三个相继的判定，所以共有 $2^3 = 8$ 种可能的组合，也就是有八种不同的路径。由图 3-8 所示可见，$PDBFQ$是最短路径，其路径的总长是 $12+2+3+4=21$。

这种判定构造是一种多步序的判定系统，驾驶员的地点位置，即图 3-7 中的各点表示系统的状态，它从给定初始状态P到给定最终状态Q经过一序列步序的判定，以使整个“代价”极小，也就是所经过的总距离最小。

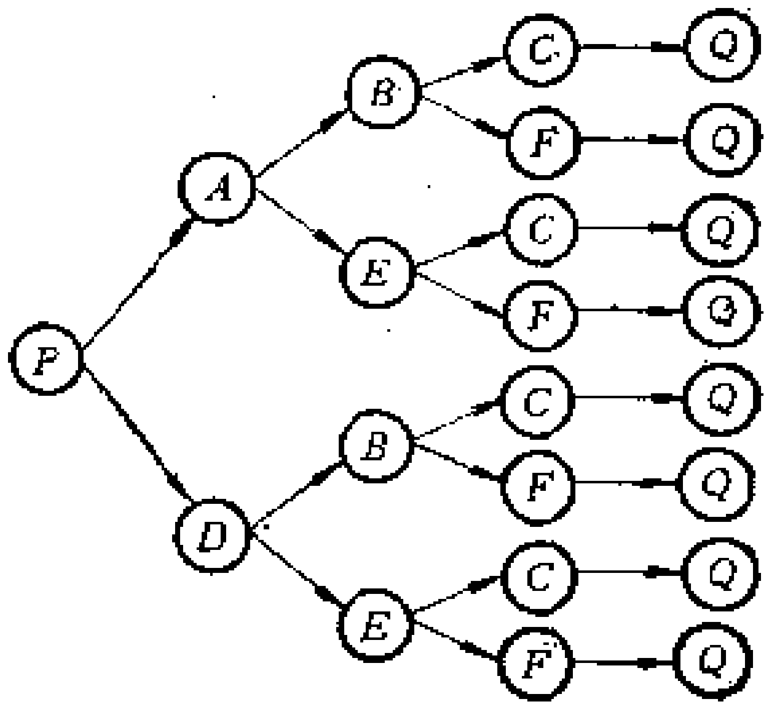


图3-8　各段路径的比较

从P点开始，也就是从初始状态$y_0=P$开始，驾驶员可选择的两种途径可用一个判定变量x_1的两个数值表出：$x_1=A$（驶到A点）和$x_1=D$(驶到D点)。如果他选定$x_1=A$，说明第一步序的输出状态是$y_1=A$，并且对应的代价r_1（P，A）就是距离$PA=10$。类似的，$x_1=D$就给出$y_1=D$，并且代价是r_1（P，D）$=12$，我们把收益在这种情况下说成代价更为恰当，当然，收益的概念本身也是相对的。照这样方式下去，问题就可以化成图 3-9 的一种四步序判定构

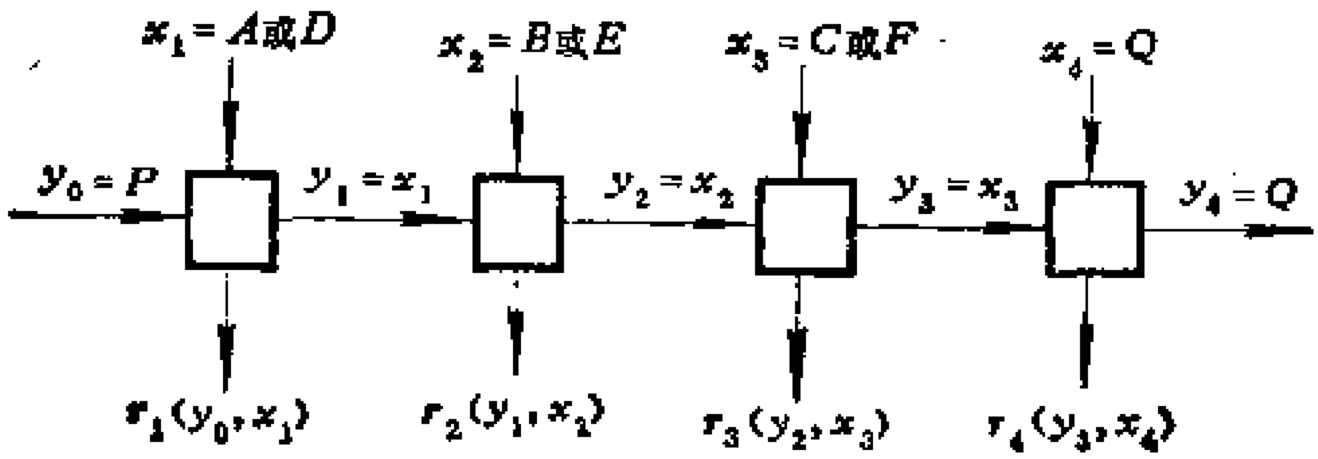


图3-9　四个步序的判定构造

造。在第四步序中没有选择，因为最终点的最后状态$y_4=Q$是给定的。

现在，变换函数是：

$$y_n=x_n \qquad n=1，2，3，4$$

代价函数是按下列顺序给出的：

$$r_n=r_n(y_{n-1}, x_n)$$

$y_0=P$,	$x_1=A$,	$r_1(y_0,\ x_1)=10$
	$x_1=D$,	$r_1(y_0,\ x_1)=12$
$y_1=A$,	$x_2=B$,	$r_2(y_1,\ x_2)=6$
	$x_2=E$,	$r_2(y_1,\ x_2)=7$
$y_1=D$,	$x_2=B$,	$r_2(y_1,\ x_2)=2$
	$x_2=E$,	$r_2(y_1,\ x_2)=2$
$y_2=B$,	$x_3=C$,	$r_3(y_2,\ x_3)=3$
	$x_3=F$,	$r_3(y_2,\ x_3)=3$
$y_2=E$,	$x_3=C$,	$r_3(y_2,\ x_3)=6$
	$x_3=F$,	$r_3(y_2,\ x_3)=4$
$y_3=C$,	$x_4=Q$,	$r_4(y_3,\ x_4)=5$
$y_3=F$,	$x_4=Q$,	$r_4(y_3,\ x_4)=4$

从 $y_0=P$ 开始，考虑到 $x_1(y_0)=A$，$x_1(y_0)=D$ 的两种情况，应进一步联系到 $y_1=x_1$ 以后，$x_2(y_1)$ 所给出的收益，我们从上列数值可以看出，应选：

$$x_1(y_0)=D$$

虽然，$r_1(y_0,\ x_1)=12$ 比较 10 大，但从 D 出发 $r_2(y_1,x_2)=2$ 比 6，7 二值加上 10 所得的 16，17 二值来得小，也就是 PDB，PDE 的径长只有 14。

当 $y_1=D$ 时，第二步序给出 $x_2=B$，$y_2=B$；如此下去，最优值的字母表示如右表。

n	y_n	x_n
0	P	
1	D	D
2	B	B
3	F	F
4	Q	Q

最优状态序列 y_0, y_1, y_2, y_3, y_4 通过网络表出的最短路径是 $PDBFQ$，而路径的全长是 $F_4(y_0)=21$。

这样的一种解法，可以用图 3-8 的一种判决树图的方法来说明。经过首三次的判决之后，驾驶员发现他自己是在 C 或 F；再向后推一步，头两次的判决使驾驶员在 B 或 F，如果他到达的是

B，他将经过C或F到达目的地Q；最好的是经过F，因为$BFQ = 3 + 4 = 7$，而$BCQ = 3 + 5 = 8$。所以，不管他怎样到达B总不要经过C，所以从B到C的一支路程可以抹去，而仅考虑从B到F。这就是贝尔曼（Bellman）最优化原理的一种应用：如果从P到Q的最优路径要经过B，则路径的未经部分，即从B到Q必须也是最优的，所以从P到Q的最优路径不会包含BCQ。

类似地，如果驾驶员经过头两次判决之后是在E点，他将要经过F，因为：

$$EFQ = 4 + 4 = 8 < ECQ = 6 + 5 = 11$$

所以，要抹去ECQ这一条路。

照这样办法，在第一次判决之后，应该抹去AE和DE，从A到Q的最短路径是：

$$ABFQ = 6 + 7 = 13$$

并且从D到Q的最短路径是：

$$DBFQ = 2 + 7 = 9$$

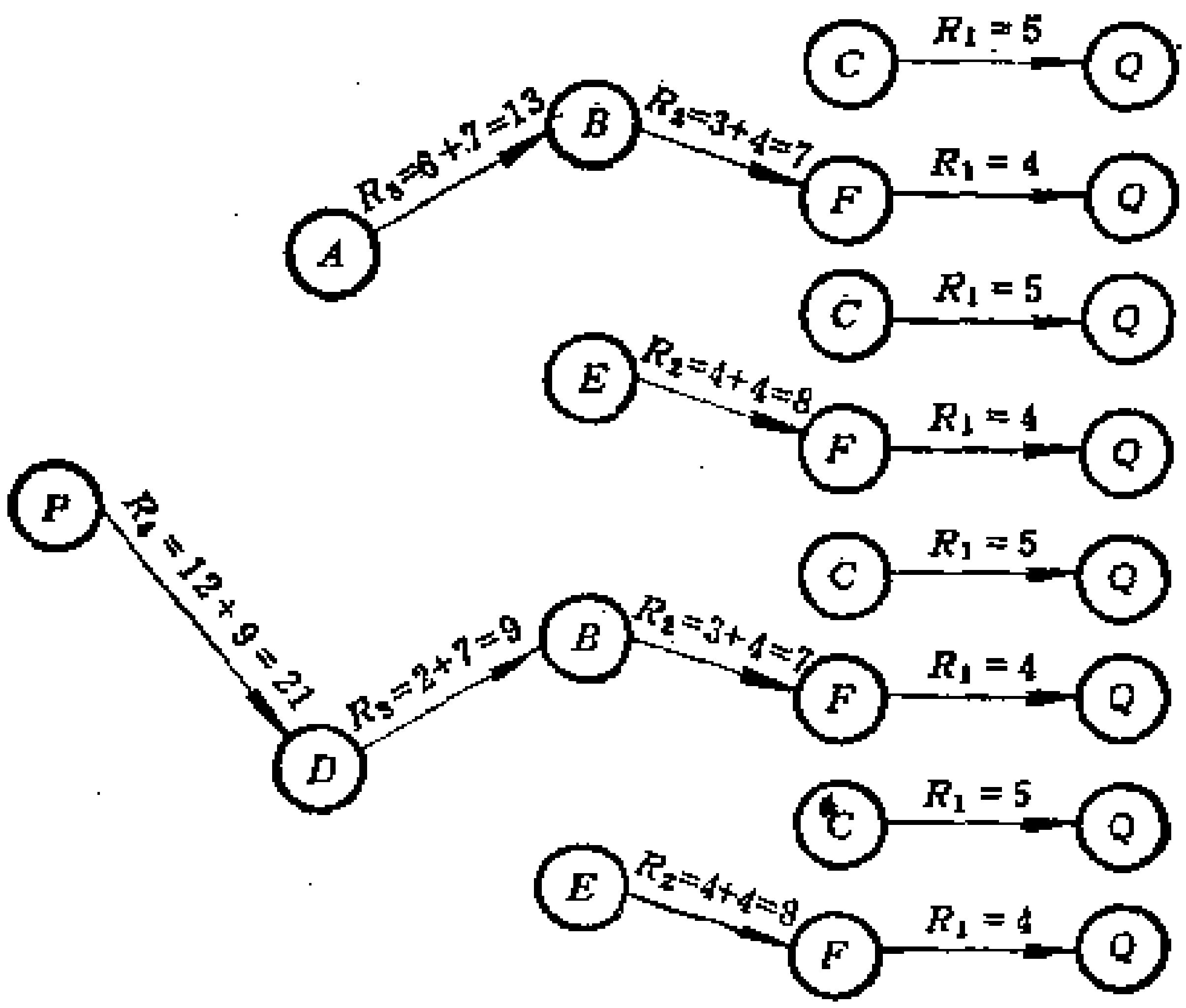


图3-10　最短路径的图解

最后，开始点要在

$$PA+ABFQ=10+13=23$$

和

$$PD+DBFQ=12+9=21$$

之间选择，所以要抹去 PA 这一条路。

在把图 3-8 中所有抹去的路段不画出时，仅有的一条从起点到终点的不断路径是 $PDBFQ$，它就表出了最短的整个路径，见图 3-10。

§5 几何规划

我们在第一章中介绍的汽轮机设计的数学模型，它们的目标函数和约束函数都是非负变量 x_1，x_2，…，x_n 的具有正系数的多项式，也就是目标函数是：

$$F(x)=\sum_{i=m_0}^{n_0} p_i(x) \tag{3-35}$$

$$p_i(x)=c_i x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_n^{a_{in}}$$

$$c_i>0, \quad x_j>0 \qquad j=1, 2, \cdots, n$$

其中：指数 a_{i1}，a_{i2}，…，a_{in} 是任意实数，并且约束条件写成规格化的形式为：

$$g_k(x)\leqslant 1 \qquad k=1, 2, \cdots, p \tag{3-36}$$

其中：

$$g_k(x)=\sum_{i=m_k}^{n_k} p_{ki}(x)$$

$$p_{ki}(x)=c_i^{(k)}\cdot x_1^{a_{i1}^{(k)}} {}^{a_{i2}^{(k)}} \cdots {}^{a_{in}^{(k)}}$$

$$k=1, 2, \cdots, p$$

式中的指数 $a_{i1}^{(k)}$，$a_{i2}^{(k)}$，…，$a_{in}^{(k)}$ 是任意实数。我们称具有式(3-35)、(3-36) 形式的数学规划问题是一种“几何规划”。

称式 (3-35)、(3-36) 是一种几何规划的原因，是我们将可

以按照算术平均极小化等价于某几何平均的极大化的对偶性质来解决这类数学规划问题。

简单地说，我们知道算术平均是大于或等于几何平均的，也就是：

$$(x_1+x_2)/2 \geqslant \sqrt{x_1x_2}, \qquad x_1>0, \quad x_2>0 \qquad (3\text{-}37)$$

1967 年，得芬（Duffin）等人利用了这种几何不等式，有效地把具有非线性约束的主问题的一个极小化要求，转化为具有线性约束的一个对偶问题。

方法是，相对于（3-35）的多项式形式的目标函数，引入任意的正的权值 δ_1，δ_2，…，$\delta_{n_0-m_0}$，或者，简单地说，相对于

$$F(\mathbf{x})=\sum_{i=1}^{k} p_i(\mathbf{x})$$

给出 δ_1，δ_2，…，δ_k，使它们具有"正规化条件"，即

$$\sum_{i=1}^{k}\delta_i=1$$

然后，把 $F(\mathbf{x})$ 写成：

$$F(\mathbf{x})=\delta_1\frac{p_1(\mathbf{x})}{\delta_1}+\delta_2\frac{p_2(\mathbf{x})}{\delta_2}+\cdots+\delta_k\frac{p_k(\mathbf{x})}{\delta_k}$$

利用（3-37）几何不等式可以有：

$$F(\mathbf{x})\geqslant\left(\frac{p_1(\mathbf{x})}{\delta_1}\right)^{\delta_1}\left(\frac{p_2(\mathbf{x})}{\delta_2}\right)^{\delta_2}\cdots\left(\frac{p_k(\mathbf{x})}{\delta_k}\right)^{\delta_K}$$

$$=\prod_{i=1}^{k}\left(\frac{c_i}{\delta_i}\right)^{\delta_i} x_1^{d_1}x_2^{d_2}\cdots x_n^{d_n} \qquad (3\text{-}38)$$

其中

$$d_i=\sum_{j=1}^{k}\delta_j a_{ji} \qquad i=1,\ 2,\ \cdots,\ n \qquad (3\text{-}39)$$

从式（3-37）容易看出等式成立的一个必要条件是 $x_1=x_2$，类似地，（3-38）不等式中等式成立的一个必要条件是对一切

$i=1, 2, \cdots, n$ 有：

$$\frac{p_i(\mathbf{x})}{\delta_i}=\text{常数} \tag{3-40}$$

现在，如果我们给出一个附加条件，使

$$d_i=\sum_{j=1}^{k}\delta_j a_{ji}=0 \tag{3-41}$$

$$i=1, 2, \cdots, n$$

则式（3-38）变成仅是 δ 的一个函数了，即

$$F(\mathbf{x})\geqslant V(\boldsymbol{\delta}), \quad \boldsymbol{\delta}\in R^k \tag{3-42}$$

其中

$$V(\boldsymbol{\delta})=\prod_{i=1}^{k}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}$$

$$\boldsymbol{\delta}=(\delta_1, \delta_2, \cdots, \delta_k)^T$$

因此，条件（3-41）称为"正交化条件"。

从式（3-42）可以看出：

$$\min_{\mathbf{x}>0} F(\mathbf{x})\geqslant\max_{\boldsymbol{\delta}\geqslant 0} V(\boldsymbol{\delta})$$

这里，对 δ 的极大化还要合于正规化条件与正交化条件，即

$$\sum_{i=1}^{k}\delta_i=1, \quad \sum_{j=1}^{k}\delta_j a_{ji}=0 \qquad i=1, 2, \cdots, n$$

假设目标函数 $F(\mathbf{x})$ 在 x^* 有极小，并且

$$F(\mathbf{x}^*)>0$$

这时，我们可以肯定有 $\boldsymbol{\delta}^*$ 使（3-42）成立等式关系。因为 $F(\mathbf{x})$ 是一个多项式，它的导数存在，若 $\mathbf{x}^*$ 是它的极小点，则有：

$$\left.\frac{\partial F}{\partial x_i}\right|_{\mathbf{x}^*}=0 \qquad i=1, 2, \cdots, n$$

或者说，由式（3-35）得：

$$x_i\left.\frac{\partial F}{\partial x_i}\right|_{\mathbf{x}^*}=\sum_{j=1}^{k}a_{ji}p_j(\mathbf{x}^*)=0 \tag{3-43}$$

$$i=1,\ 2,\ \cdots,\ n$$

如果定义

$$\delta_i^* = p_i(\mathbf{x}^*)/F(\mathbf{x}^*) \tag{3-44}$$

$$i=1,\ 2,\ \cdots,\ k$$

式（3-44）显然合于正规化条件

$$\sum_{i=1}^{k} \delta_i^* = 1$$

而且直接从式（3-43）可以看出正交化条件是满足的。再从 δ_i^* 的定义，还有以下关系式：

$$(F(\mathbf{x}^*))^{\delta_i^*} = \left(\frac{p_i(\mathbf{x}^*)}{\delta_i^*}\right)^{\delta_i^*}$$

或者

$$\prod_{i=1}^{k}(F(\mathbf{x}^*))^{\delta_i^*} = \prod_{i=1}^{k}\left(\frac{p_i(\mathbf{x}^*)}{\delta_i^*}\right)^{\delta_i^*} \tag{3-45}$$

上式的左边是合于正规化条件的 $F(\mathbf{x}^*)$，右边是合于正交化条件的 δ^* 的函数 $V(\delta^*)$，所以建立了下式：

$$F(\mathbf{x}^*) = V(\delta^*) \tag{3-46}$$

这就是我们所希望得到的对偶关系。一旦关于 $V(\delta)$ 合于约束条件的极大问题解决了，极小化 $F(\mathbf{x})$ 的对偶问题就解决，从而解决了极小化 $F(\mathbf{x})$ 的重要问题。

因为，在解 $V(\delta)$ 使其极大以后得到 $\delta^*\geqslant 0$ 以后，从式（3-44）可以解出 $\mathbf{x}^*$，即

$$\delta_i^* = p_i(\mathbf{x}^*)/F(\mathbf{x}^*)$$

并且从式（3-35）可知，上式中 $i=m_0,\ m_0+1,\ \cdots,\ n_0$。

而在式（3-35）中有：

$$p_i(\mathbf{x}) = c_i x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_n^{a_{in}}$$

$$c_i > 0,\quad x_j > 0 \qquad j=1,\ 2,\ \cdots,\ n$$

式（3-46）又给出：

$$F(\mathbf{x}^*) = V(\delta^*)$$

所以有：

$$V(\delta^*)\delta_i^* = c_i x_1^{*a_{i1}} x_2^{*a_{i2}} \cdots x_n^{*a_{in}}$$

取对数得：

$$\ln\left[V(\delta^*)\frac{\delta_i^*}{c_i}\right] = \sum_{j=1}^{n} a_{ij}(\ln x_j^*) \qquad (3\text{-}47)$$

$$i = m_0,\ m_0+1,\ \cdots,\ n_0$$

解出式（3-47）中的 x^* 就是我们使用几何规划求解的目的。

例 7　极小化

$$F(x) = ax^2 + \frac{b}{x^2} \qquad a,\ b > 0$$

它是一个无约束的一维的几何规划形式，很容易看出，由式（3-37）就有：

$$F(x) \geqslant \left(\frac{ax^2}{1/2}\right)^{1/2}\left(\frac{b}{x^2/2}\right)^{1/2} = 2\sqrt{ab}$$

也就是 $\delta_1^* = \delta_2^* = 1/2$，等式在

$$x^2 = \sqrt{b/a}$$

时取为：

$$\min_x F(x) = F[(b/a)^{1/4}] = 2(ab)^{1/2}$$

例 8　极小化

$$F(x) = F(x_1,\ x_2) = \frac{a}{x_1 x_2} + \frac{bx_1}{x_2} + cx_2$$

$$a > 0,\ b > 0,\ c > 0$$

是一个无约束二维几何规划问题，从而有：

$$V(\delta_1,\ \delta_2,\ \delta_3) = \left(\frac{a}{\delta_1}\right)^{\delta_1}\left(\frac{b}{\delta_2}\right)^{\delta_2}\left(\frac{c}{\delta_3}\right)^{\delta_3}$$

待极大化后，并使

$$\delta_1 + \delta_2 + \delta_3 = 1,\ d_1 = -\delta_1 + \delta_2 = 0,$$

$$d_2 = -\delta_1 - \delta_2 + \delta_3 = 0$$

对 δ 求解得：

$$\delta_1^* = \frac{1}{4},\quad \delta_2^* = \frac{1}{4},\quad \delta_3^* = \frac{1}{2}$$

并且

$$V(\delta^*) = (4a)^{1/4}(4b)^{1/4}(2c)^{1/2} = 2^{3/2}a^{1/4}b^{1/4}c^{1/2}$$

再从

$$\delta_i^* F(\mathbf{x}^*) = p_i(x^*) \Rightarrow \delta_i^* V(\delta^*) = p_i(\mathbf{x}^*)$$

就有

$$cx_2^* = \frac{V(\delta^*)}{2}$$

$$\frac{bx_1^*}{x_2^*} = \frac{V(\delta^*)}{4}$$

所以，最优化点是：

$$x_1^* = \sqrt{\frac{a}{b}}$$

$$x_2^* = 2^{1/2}a^{1/4}b^{1/4}c^{-1/2}$$

一般地来说，几何规划具有的形式是式（3-35）、（3-36）的统一写法，即

$$\begin{aligned} &\text{极小化} && g_0(\mathbf{x}) \\ &\text{使合于} && g_k(\mathbf{x}) \leqslant 1 \qquad k = 1,\ 2,\ \cdots,\ p \\ &&& x = (x_1,\ x_2,\ \cdots,\ x_n)^\tau > 0 \end{aligned} \tag{3-48}$$

其中

$$g_k(\mathbf{x}) = \sum_{i=m_k}^{n_k} c_i \prod_{j=1}^{n} x_j^{a_{ij}}$$

$$k = 0,\ 1,\ 2,\ \cdots,\ p$$

$c_i > 0$，m_k 和 n_k 是用来表示相继顺序项的个数，例如：$m_0 = 1,\ 2,\ \cdots,\ n_0$ 表示目标函数的项数，而 $n_p = m$ 表示包括目标函数，约束函数所有项数的总项数。

按上述方法，对应于式（3-48）的对偶几何规划是：

$$\text{极大化} \qquad V(\delta) = \prod_{i=1}^{m} (c_i/\delta_i)^{\delta_i} \prod_{k=1}^{p} \lambda_k^{\lambda_k} \tag{3-49}$$

使合于

$$\sum_{i=1}^{n_0}\delta_i=1$$

$$\sum_{i=1}^{m}a_{ij}\delta_i=0\qquad j=1,\ 2,\ \cdots,\ n\tag{3-50}$$

$$\delta_i\geqslant 0\qquad i=1,\ 2,\ \cdots,\ m$$

其中

$$\lambda_k=\sum_{i=m_k}^{n_k}\delta_i\qquad k=1,\ 2,\ \cdots,\ p\tag{3-51}$$

由于式（3-50）、（3-51）解的形式是：

$$\delta_i=b^{(0)}+\sum_{j=1}^{d}r_j b_i^{(j)}\qquad i=1,\ 2,\ \cdots,\ m$$

所以，这里新的独立变量r_j($j=1,\ 2,\ \cdots,\ d$）是使$\delta\geqslant 0$的数值，而且$d=m-n-1$，$b^{(0)}$，$b^{(j)}$($j=1,\ 2,\ \cdots,\ d$）是常数。

第四章　有约束函数的极小化准则

§1　等式约束问题

在讨论有约束的极小化问题中，很重要的一个方面是应该弄清楚拉格朗日乘子的作用。拉格朗日乘子在微积分中是针对带有边端条件的极值问题引用的，这时，边端条件是用一些等式约束给出，例如：

$$h_i(\mathbf{x})=0 \qquad i=1,\ 2,\ \cdots,\ l,\ \mathbf{x}\in R^n \tag{4-1}$$

就简单的极小化问题来说，如果要求

$$\min_{x_1,x_2,x_3} F(x_1,\ x_2,\ x_3)$$

使合于约束条件

$$h_1(x_1,\ x_2,\ x_3)=0,\ h_2(x_1,\ x_2,\ x_3)=0 \tag{4-2}$$

这时，可行域 S 是由 h_1 和 h_2 隐式所确定的 R^3 中两个曲面交成的一条曲线。设问题在 S 中有极小点 $\mathbf{x}^*=(x_1^*,\ x_2^*,\ x_3^*)^T$存在，并且 F，h_1，h_2 在包含 $\mathbf{x}^*$在内的一个开集上都有连续的一阶导数，而且梯度向量 ∇h_1 和 ∇h_2 是线性独立的，就可以有两个不全为零的拉格朗日乘子 λ_1，λ_2 使

$$\nabla F(\mathbf{x}^*)+\lambda_1\nabla h_1(\mathbf{x}^*)+\lambda_2\nabla h_2(\mathbf{x}^*)=0 \tag{4-3}$$

成立。由此，从式（4-3）的三个方程和（4-2）两个方程就可以解出 λ_1，λ_2 和 $\mathbf{x}^*$。

一般地说，设 $F(\mathbf{x})$，$h_i(\mathbf{x})$ （$i=1,\ 2,\ \cdots,\ l \quad l<n$）是在 R^n 中的实值函数，如果它们的一阶偏导数在 R^n 中的一个开集 T 上连续，若 S 是 T 的子集，$h_i(\mathbf{x})$（$i=1,\ 2,\ \cdots,\ l$）在 S 中都是零，即

$$S=\{\mathbf{x}|x\in T,\ h_i(\mathbf{x})=0 \qquad i=1,\ 2,\ \cdots,\ l\} \tag{4-4}$$

并且问题在 S 中有极小点 $\mathbf{x}^*=(x_1^*,\ x_2^*,\ \cdots,\ x_n^*)^T$，同时有 $\mathbf{x}^*$的一个邻域 $N(\mathbf{x}^*)\subset S$，对于 $N(\mathbf{x}^*)$ 中任何一点 $\mathbf{x}$ 有：

$$F(\mathbf{x}^*) \leqslant f(\mathbf{x}) \tag{4-5}$$

若约束函数 $h_i(\mathbf{x})$（$i=1,\ 2,\ \cdots,\ l$）的雅可比阵

$$\frac{\partial(h_1,\ h_2,\ \cdots,\ h_l)}{\partial(x_1,\ x_2,\cdots,x_n)}=\begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ & \cdots\cdots & & \\ \frac{\partial h_l}{\partial x_1} & \frac{\partial h_l}{\partial x_2} & \cdots & \frac{\partial h_l}{\partial x_n} \end{pmatrix} \tag{4-6}$$

的秩在 $\mathbf{x}^*$ 是 l，则有 l 个不全为零的实数 $\lambda_1,\ \lambda_2,\ \cdots,\ \lambda_l$ 使

$$\nabla F(\mathbf{x}^*)+\sum_{i=1}^{l}\lambda_i \nabla h_i(\mathbf{x}^*)=0 \tag{4-7}$$

事实上，在上述条件下，总可以从式（4-6）中取出 l 阶行列式，设其首主子行列式不为零，由隐函数组存在定理可知，能解出显式，且设前 l 个变量是：

$$\begin{aligned} x_1&=\varphi_1(x_{l+1},\ x_{l+2},\ \cdots,\ x_n) \\ x_2&=\varphi_2(x_{l+1},\ x_{l+2},\ \cdots,\ x_n) \\ &\cdots\cdots \\ x_l&=\varphi_l(x_{l+1},\ x_{l+2},\ \cdots,\ x_n) \end{aligned} \tag{4-8}$$

这样，就可以把函数 $F(\mathbf{x})$ 求极小值的问题化为无约束求极小值问题，即

$$\min_{(x_{l+1},x_{l+2},\cdots,x_n)} F(\varphi_1(x_{l+1},\cdots,x_n),\varphi_2(x_{l+1},\cdots,x_n),\cdots\varphi_l(x_{l+1},\cdots,x_n), x_{l+1},\cdots,x_n) \tag{4-9}$$

现在，函数 $F(x_1,\ x_2,\ \cdots,\ x_n)$ 在 $\mathbf{x}^*$ 有相对于给定约束条件的极小值，或者说复合函数

$$F(\varphi_1(x_{l+1},\cdots,x_n),\varphi_2(x_{l+1},\ \cdots,x_n),\cdots,\varphi_l(x_{l+1},\cdots,x_n), x_{l+1},\cdots,x_n) \tag{4-10}$$

在 $\mathbf{x}^*$ 有绝对极小值，从而对于自变量 $(x_{l+1},\ \cdots,\ x_n)$ 的微分

(dx_{l+1}, dx_{l+2}, …, dx_n) 所形成的函数微分应该在 x^* 是零，再从微分形式不变性来说，可以写成：

$$\sum_{j=1}^{n} \frac{\partial F}{\partial x_j} dx_j = 0 \tag{4-11}$$

其中：dx_1, dx_2, …, dx_l 可理解为显函数（4-8）在 x^*的微分，并且（4-11）式当然并不能断定诸微分前面的系数都等于零，因为这些微分并非全是可以任意的。要想把各应变量的微分 dx_1, dx_2, …, dx_l 去掉，只要针对等式约束（4-1）求其全微分，则有：

$$\sum_{j=1}^{n} \frac{\partial h_i}{\partial x_j} dx_j = 0 \qquad i=1,\ 2,\ \cdots,\ l \tag{4-12}$$

从式（4-12）可知 dx_1, dx_2, …, dx_l可用dx_{l+1}, …, dx_n 表出，代入式(4-11)就可以消除应变量的微分。

拉格朗日乘子的思想是把自变量 x_{l+1}, x_{l+2}, …, x_n 和应变量 x_1, x_2, …, x_l 保持着同样的地位，暂时用 l 个乘数 λ_i($i=1, 2, \cdots, l$)，依次乘（4-12）各式，然后把所得结果和（4-11）相加得等式：

$$\sum_{j=1}^{n}\left(\frac{\partial F}{\partial x_j} + \lambda_1\frac{\partial h_1}{\partial x_j} + \lambda_2\frac{\partial h_2}{\partial x_j} + \cdots + \lambda_l\frac{\partial h_l}{\partial x_j}\right) dx_j = 0 \tag{4-13}$$

式中：dx_1, dx_2, …, dx_l 仍表示显式（4-8）所给出的隐函数 φ_1, φ_2, …, φ_l 的微分，并且一切导数都是在极小点 x^*取值。

现在选择乘数 $\lambda_i=\lambda_i^0$($i=1, 2, \cdots, l$) 使应变量微分 dx_1, dx_2, …, dx_l 前的系数都等于零，即

$$\frac{\partial F}{\partial x_j} + \lambda_1^0\frac{\partial h_1}{\partial x_j} + \lambda_2^0\frac{\partial h_2}{\partial x_j} + \cdots + \lambda_l^0\frac{\partial h_l}{\partial x_j} = 0 \tag{4-14}$$

$$j=1,\ 2,\ \cdots,\ l$$

这是可以做到的，因为已经知道 λ_1^0, λ_2^0, …, λ_l^0的系数行列式不为零。

在选定了这些乘数 λ_1^0, λ_2^0, …, λ_l^0 之后，等式(4-13)成为：

$$\sum_{j=l+1}^{n}\left(\frac{\partial F}{\partial x_j}+\lambda_1^0\frac{\partial h_1}{\partial x_j}+\lambda_2^0\frac{\partial h_2}{\partial x_j}+\cdots+\lambda_l^0\frac{\partial h_l}{\partial x_j}\right)dx_j=0$$

(4-15)

其中：所有的微分 dx_{l+1}, dx_{l+2}, …, dx_n 都是自变量微分，因此，所有这些微分前的系数必须都是零，也就是和式（4-14）并列的又有：

$$\frac{\partial F}{\partial x_j}+\lambda_1^0\frac{\partial h_1}{\partial x_j}+\lambda_2^0\frac{\partial h_2}{\partial x_j}+\cdots+\lambda_l^0\frac{\partial h_l}{\partial x_j}=0 \qquad (4\text{-}16)$$

$$j=l+1,\ l+2,\ \cdots,\ n$$

这样，从式（4-14）、（4-16）这 n 个方程和 l 个（4-1）约束方程就能解出 n 个未知量 x_1^*, x_2^*, …, x_n^* 以及 l 个乘数 λ_1^0, λ_2^0, …, λ_l^0。

所以，在解有等式约束的函数极小化问题中，我们可以采用拉格朗日乘数 λ_1, λ_2, …, λ_l 引入辅助函数，即

$$\Phi=F+\lambda_1h_1+\lambda_2h_2+\cdots+\lambda_lh_l \qquad (4\text{-}17)$$

因此式（4-14）、（4-16）就可以写成：

$$\frac{\partial\Phi}{\partial x_j}=0 \qquad j=1,2,\cdots,n \qquad (4\text{-}18)$$

用式（4-18）和式（4-1）联解出 $\mathbf{x}^*$ 就是拉格朗日乘子法对极值存在的必要条件。

式（4-18）就式(4-2)的问题来说就是式(4-3)，它说明在极小点 $\mathbf{x}^*$ 的梯度向量 $\nabla F(\mathbf{x}^*)$ 是落在约束函数梯度向量 $\nabla h_1(\mathbf{x}^*)$ 和 $\nabla h_2(\mathbf{x}^*)$ 所决定的平面，这个平面在 $\mathbf{x}^*$ 点垂直于 $h_1(\mathbf{x})=0$, $h_2(\mathbf{x})=0$ 曲线，见图 4-1。还注意到，这时 $F(\mathbf{x})$ 即是沿着 S 曲线在曲线的 $\mathbf{x}^*$ 点上达到极值，$F(\mathbf{x})$ 沿 S 上的方向导数应当是零，也就是，若

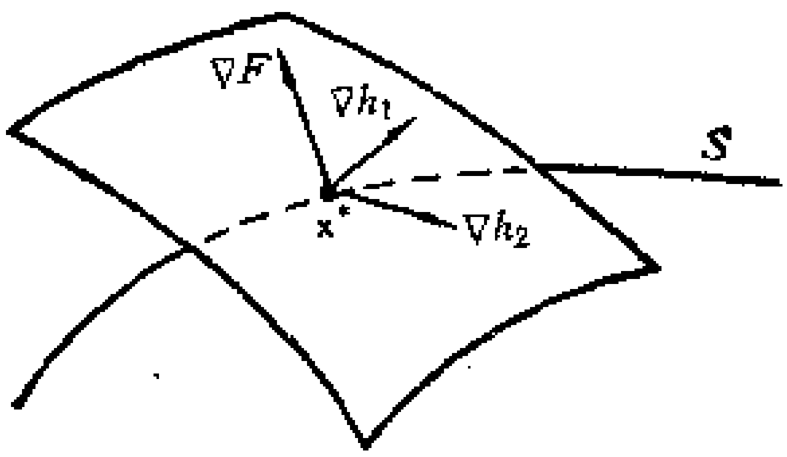


图4-1 $n=3$，$l=2$ 的有约束极小化

$$d\mathbf{x}=(dx_1,\ dx_2,\ dx_3)^T$$

表示曲线S上的一段无穷小线元，则

$$(\nabla F(\mathbf{x}^*),\ d\mathbf{x})=0$$

是式（4-11）的一种特例，同时式（4-12）给出了两个切平面方程，即

$$(\nabla h_1(\mathbf{x}^*),\ d\mathbf{x})=0$$
$$(\nabla h_2(\mathbf{x}^*),\ d\mathbf{x})=0$$

并决定了S在$\mathbf{x}^*$的切线。

如果极小化函数$F(\mathbf{x})$的约束条件是线性等式约束，即

$$\mathbf{A}\mathbf{x}=\mathbf{b}(\mathbf{A}\text{ 是 } l\times n \text{ 矩阵，} l<n) \tag{4-19}$$

并且$\mathbf{A}$的秩是l。若$F(\mathbf{x})$有一阶连续偏导数，我们就可按上述准则，用拉格朗日乘子给出有约束极小化问题的必要条件，其办法是构造拉格朗日函数（4-17），函数为：

$$\Phi(\mathbf{x})=F(\mathbf{x})+(\lambda,\mathbf{A}\mathbf{x}-\mathbf{b}) \tag{4-20}$$

其中：λ是一个l维向量，那末，必要条件是：

$$\nabla\Phi(\mathbf{x}^*)=\nabla F(\mathbf{x}^*)+\mathbf{A}^T\lambda=0 \tag{4-21}$$

同时

$$(\lambda,\mathbf{A}\mathbf{x}^*-\mathbf{b})=0 \tag{4-22}$$

关于（4-21）式，我们可以不采用隐函数存在定理，在约束函数是凸的或线性的定义在R^n中的一个凸域时，由可行方向的移动作用直接判定极小点的必要条件和拉格朗日乘数的存在。因为，对于充分小的$t>0$，总可就一个可行方向$\mathbf{d}\in R^n$使$\mathbf{x}^*+t\mathbf{d}$是可行的点，从而将$F(\mathbf{x})$在$\mathbf{x}^*$展成：

$$F(\mathbf{x}^*+t\mathbf{d})=F(\mathbf{x}^*)+t(\nabla F(\mathbf{x}^*),\mathbf{d})+o(t) \tag{4-23}$$

其中：$o(t)$是当t趋于零关于t的高阶无穷小；并且使$\mathbf{A}(\mathbf{x}^*+t\mathbf{d})=\mathbf{b}$，也就是：

$$\mathbf{A}\mathbf{d}=0 \tag{4-24}$$

从$\mathbf{x}^*$是有约束的极小点这个事实，我们就知道对于所有使$\mathbf{A}\mathbf{d}=0$的$\mathbf{d}$将有：

$$(\nabla F(\mathbf{x}^*),\ \mathbf{d})\geqslant 0$$

也就是在 R^n 中不存在d能使

$$(\nabla F(\mathbf{x}^*),\ \mathbf{d})<0 \tag{4-25}$$

$$\mathbf{A}\mathbf{d}=0$$

同时成立。

今在 R^{l+1} 中定义一个集合 V，它是：

$$V=\left\{\begin{pmatrix}v_0\\ \mathbf{v}\end{pmatrix}\in R^{l+1},\ (\nabla F(\mathbf{x}^*),\ \mathbf{d})<v_0,\right.$$
$$\left.\mathbf{A}\mathbf{d}=\mathbf{v}\in R^l \text{ 对 } R^n \text{ 中某 } \mathbf{d} \text{ 成立}\right\} \tag{4-26}$$

1957年由樊畿等人● 给出的凸函数性质知道，这个 V 集合是非空的凸集，也就是对于任何给定的 $\mathbf{x}^1$，$\mathbf{x}^2\in V$，对于所有的 $0\leqslant\mu\leqslant 1$ 构成的点

$$\mathbf{x}=\mu\mathbf{x}^1+(1-\mu)\mathbf{x}^2 \tag{4-27}$$

都是 V 中的点。现在 R^{l+1} 中的原点不在 V 中，所以，总可以找到 λ_0 纯量和 l 向量 $\boldsymbol{\lambda}$ 对于 V 中所有的 $\begin{pmatrix}v_0\\ \mathbf{v}\end{pmatrix}$ 恒有：

$$\lambda_0 v_0+(\boldsymbol{\lambda},\mathbf{v})>0 \tag{4-28}$$

并且 $\lambda_0\geqslant 0$，$\boldsymbol{\lambda}$ 的所有分量都不小于零。

就 R^n 中所有的 $\mathbf{d}$，给定一个充分小的 $\varepsilon>0$，使：

$$\begin{pmatrix}v_0\\ \mathbf{v}\end{pmatrix}=\begin{pmatrix}(\nabla F(\mathbf{x}^*),\ \mathbf{d})\\ \mathbf{A}\mathbf{d}\end{pmatrix}+\varepsilon\begin{pmatrix}1\\ 0\end{pmatrix}$$

在 V 中，由于 ε 是任意的，所以从式(4-28)，我们得到对所有 $\mathbf{d}$ 的(4-28)式左边的下限为：

$$\operatorname*{INF}_{\mathbf{d}}\left[\lambda_0(\nabla F(\mathbf{x}^*),\mathbf{d})+(\boldsymbol{\lambda},\mathbf{A}\mathbf{d})\right]\geqslant 0$$

也就是

$$(\lambda_0\nabla F(\mathbf{x}^*)+\mathbf{A}^T\boldsymbol{\lambda},\ \mathbf{d})\geqslant 0$$

对一切的 $\mathbf{d}\in R^n$ 成立，所以只能是：

$$\lambda_0\nabla F(\mathbf{x}^*)+\mathbf{A}^T\boldsymbol{\lambda}=0 \tag{4-29}$$

由设定，$\mathbf{A}$ 的秩是 l，$\mathbf{A}$ 的各行向量就应是线性独立的，所以必

● 见：Fan K., I. Glicksburg, and A. J. Hoffman,《Systems of Inequalities Involving Convex Functions》, Proc. Amer. Math. Soc., 8 (1957) 617-22.

然是 $\lambda_0 \neq 0$，从而式(4-29)就是式(4-21)的结果。

§2 线性不等式约束问题

我们现在来分析带有一组不等式约束的极小化问题，即

$$极小化 F(\mathbf{x}),\quad \mathbf{x}\in R^n$$

使合于

$$g_i(\mathbf{x})\leqslant 0 \qquad i=1, 2, \cdots, m \tag{4-30}$$

首先，在这一节，我们先就所有的函数 $g_i(\mathbf{x})$ 都是线性的特殊情况来加以讨论，也就是，对于线性不等式约束的极小化问题，来分析一下拉格朗日乘数的作用。

值得提出的是在等式约束的极小化问题中，在一定条件下，拉格朗日乘数的作用是在极小点 $\mathbf{x}^*$ 使问题成立的必要条件为：

$$\nabla F(\mathbf{x}^*)+\sum_{i=1}^{l}\lambda_i\nabla h_i(\mathbf{x}^*)=0 \tag{4-7}$$

并且由于等式约束 $h_i(\mathbf{x})=0$ ($i=1, 2, \cdots, l$) 所以就有：

$$\sum_{i=1}^{l}\lambda_i\nabla h_i(\mathbf{x}^*)=0 \tag{4-31}$$

从而(4-17)定义的拉格朗日补助函数就给出：

$$\Phi(\mathbf{x}^*)=F(\mathbf{x}^*) \tag{4-32}$$

一般地说，关于不等式约束的极小化问题，拉格朗日乘数的作用就将要比较式(4-7)、(4-31)的关系减弱。

线性不等式约束的极小化问题虽然是不等式约束极小化问题中的一种很特殊的类型，但是因为很多关于线性规划的理论都能应用到这一类问题中来，所以就显得特别重要。

现在，如果极小化问题是

$$\min F(\mathbf{x}),\quad \mathbf{x}\in R^n$$

使合于

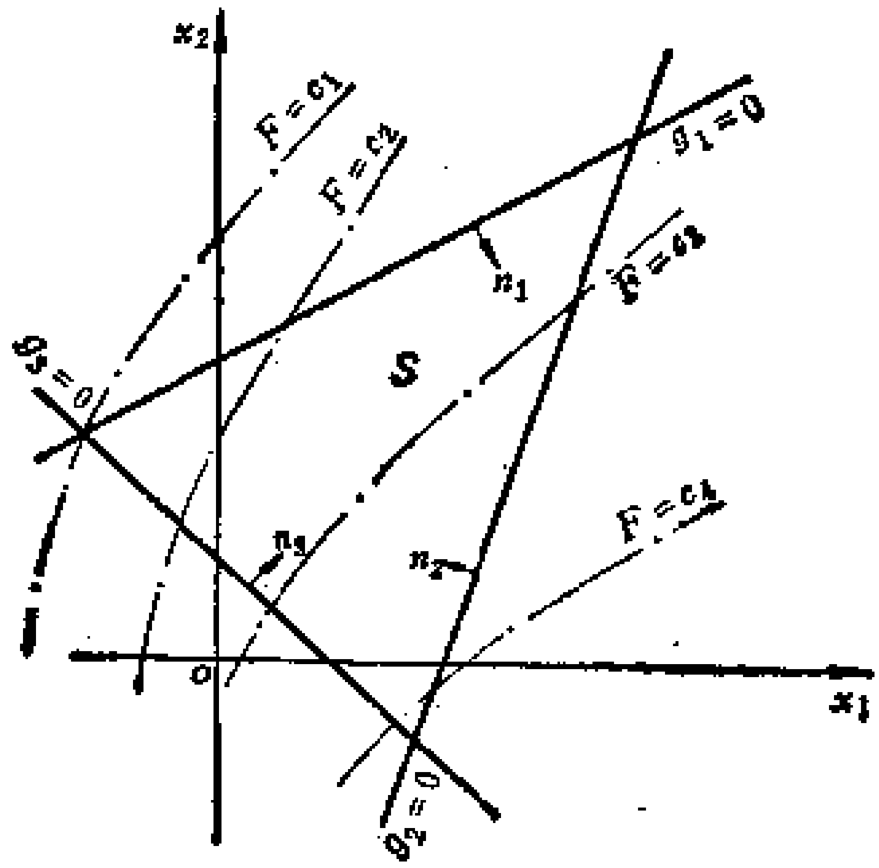
$$g_i(\mathbf{x})=-(\mathbf{u}^{(i)}, \mathbf{x})+b_i\leqslant 0 \qquad i=1, 2, \cdots, m \tag{4-33}$$

这里，每一个约束方程在 R^n 中确定了一个半空间，可行域就是由这 m 个方程所划出的 m 个半空间的交所构成的 S，它是 R^n 中的一个多面体。向量 $\mathbf{u}^{(i)}$ 就是由 $g_i(\mathbf{x})=0$ 所决定的超平面的法向，式(4-33)中的负号就是表示所选的 $\mathbf{u}^{(i)}$ 是指向 S 内部的。不失一般性，我们可以设定 $\mathbf{u}^{(i)}$，$i=1, 2, \cdots, m$ 是 m 个线性独立向量，因为，相反的情况时，我们总可以事先把式(4-33)归并为 $\mathbf{u}^{(i)}$ 是线性独立的情况。

设 $\mathbf{x}^*$ 是有约束的极小点，我们在这里定义"起作用的约束"附标集是：

$$\mathrm{I}=\{\,i\,|g_i(\mathbf{x}^*)=0\,\} \tag{4-34}$$

由于 $\mathbf{x}^*$ 总是被一些不等式约束中的不等式关系满足同时又被一些不等式约束中的等式关系满足，I 集合总是非空的。如图 4-2 就是 $n=2$，$m=3$，$\mathrm{I}=\{1, 3\}$的情况。



$c_1<c_2<c_3<c_4$

图4-2 线性约束的内向向量

如果在可行域 S 中任取一个可行点 $\mathbf{x}$，则向量 $\mathbf{x}-\mathbf{x}^*$ 是从 $\mathbf{x}^*$ 指向 S 的，是一种内向向量，由于 $\mathbf{u}^{(i)}=-\nabla g_i(\mathbf{x}^*)$，所以这个内向向量可以如下刻划：

对一切 $i\in I$ 和一切 $\mathbf{x}\in S$ 有：

$$(\mathbf{u}^{(i)},\ \mathbf{x}-\mathbf{x}^*)\geqslant 0 \tag{4-35}$$

或

$$(\nabla g_i(\mathbf{x}^*),\ \mathbf{x}-\mathbf{x}^*)\leqslant 0 \tag{4-36}$$

也就是，一个内向向量等价于从 $\mathbf{x}^*$ 移向的一个可行方向。又因为 $F(\mathbf{x})$ 是在 $\mathbf{x}^*$ 极小化的，所以对于可行域中的方向导数必然是不小于零，也就是：

$$(\nabla F(\mathbf{x}^*),\ \mathbf{x}-\mathbf{x}^*)\geqslant 0 \tag{4-37}$$

这样，我们就可以按上一节提到的可行方向那样来进行分析，但是，在这里我们将用另一种等价的方法来加以说明。

首先，我们先介绍以下内容：

法卡(Farkas)引理

设给出 $\mathbf{b}\in R^n$ 和一个 $m\times n$ 矩阵 $\mathbf{A}$。假设有一个非零向量 $\mathbf{x}$ 使 $\mathbf{A}\mathbf{x}\geqslant 0$。则当且仅当有某 $\boldsymbol{\lambda}\geqslant 0$（即 $\boldsymbol{\lambda}$ 的每一分量不小于零）使

$$\mathbf{b}=\mathbf{A}^T\boldsymbol{\lambda} \tag{4-38}$$

时，就对一切合于 $\mathbf{A}\mathbf{x}\geqslant 0$ 的 $\mathbf{x}$ 有：

$$(\mathbf{b},\mathbf{x})\geqslant 0 \tag{4-39}$$

要说明并证明这个引理，我们可以用 $\mathbf{A}$ 的行向量写出：

$$\mathbf{A}^T=(\mathbf{a}^1,\mathbf{a}^2,\cdots,\mathbf{a}^m)$$

则法卡引理说明对一切使

$$(\mathbf{a}^i,\mathbf{x})\geqslant 0\qquad i=1,2,\cdots,m$$

就有

$$(\mathbf{b},\mathbf{x})\geqslant 0$$

的结果的充分而必要的条件是式(4-38)。而式(4-38)又可以更清楚的写成：

$$\mathbf{b}=\sum_{i=1}^{m}\lambda_i\mathbf{a}^i,\quad \lambda_i\geqslant 0$$

$$i=1,2,\cdots,m$$

从几何意义上来说，这就表示 b 是在 a^i，$i=1,2,\cdots,m$ 所张开的凸多面锥 K 之中，见图 4-3。而使 $Ax\geqslant 0$ 关系成立的 x 的存在性就说明了所有的 a^i 必须指向正交于 x 的超平面 H 的同一边，所以锥形如图 4-3 所示的形状。

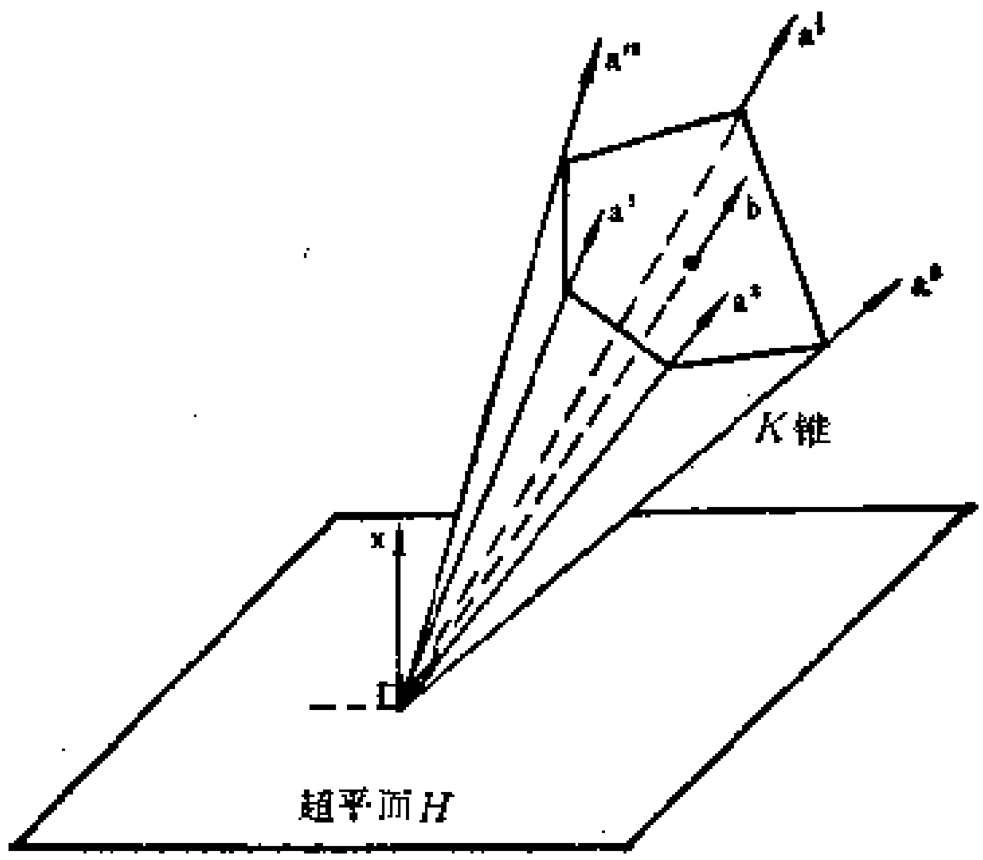


图4-3　由 $a^i(i=1,2,\cdots,m)$ 张开的多面体和 b 之间的几何关系

引理指出，若有 $\lambda\geqslant 0$ 使 $b=A^T\lambda$，则对一切满足 $Ax\geqslant 0$ 的 x 就有 $b^Tx=\lambda^TAx\geqslant 0$。

现在来反证这个引理，如果式(4-38)不成立，则 b 就不在 K 中，从而就可以考虑两种情况：

1）在 K 中最接近于从原点到 b 的顶端的点是多面凸锥上的一个面或者是 K 的一条边。

2）原点是最接近的点。

但是，适当的投影就可以把 K 和 b 画成图 4-4 中(a)的形状和(b)的形状。二者都可以选出一个向量 ξ 使：

$$A\xi\geqslant 0 \quad 同时(b,\xi)\leqslant 0$$

这就和假设矛盾。

法卡引理还可以等价地说：给定 $b\in R^n$ 和一个 $m\times n$ 矩阵 A

以后，或者是：

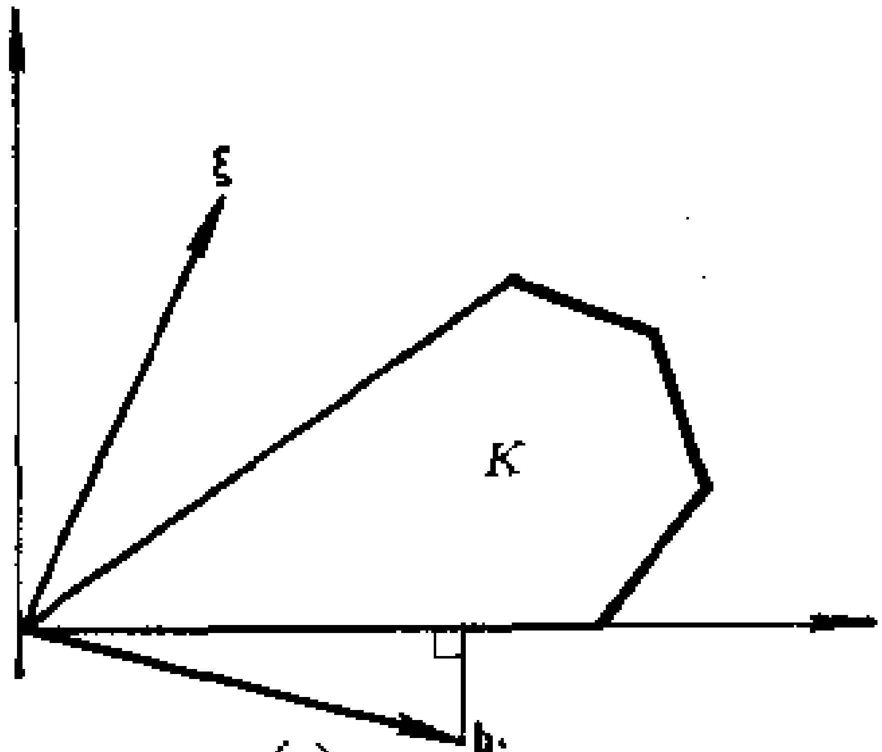
$$Ax \leqslant 0, \quad b^T x > 0$$

有一个解 $x \in R^n$，或者是：

$$b = A^T \lambda, \quad \lambda \geqslant 0$$

有一个解 $\lambda \in R^m$，但决不会二者同时成立。

现在，我们可以对一般线性不等式约束的极小化问题(4-33)来引用法卡引理的结论了。式(4-35)、(4-36)和(4-37)就说明有



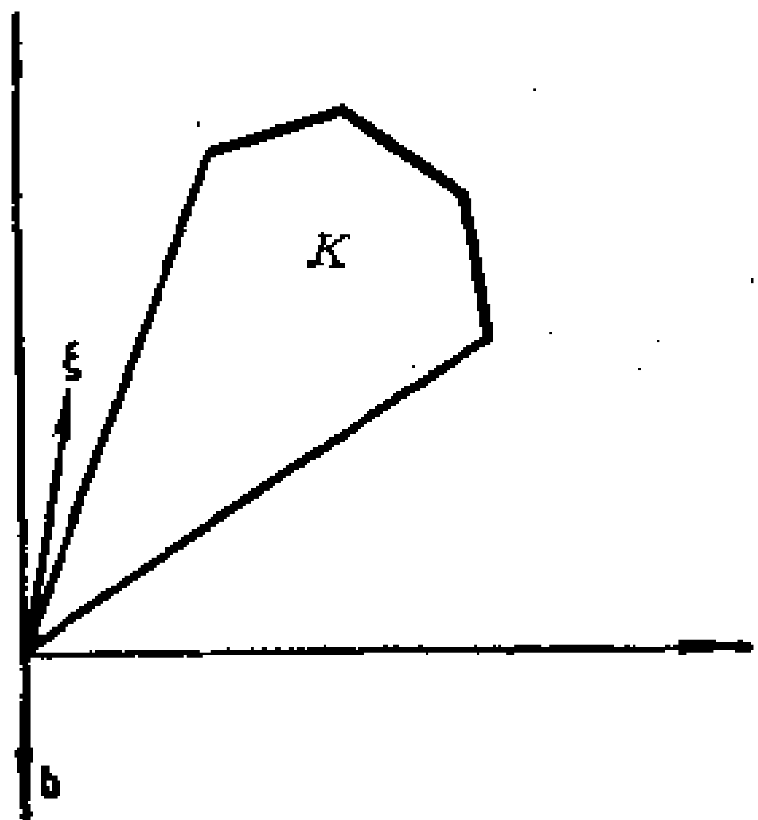


图4-4 多面锥和 b 的投影

一组非负纯量常数 λ_i 使：

$$\nabla F(\mathbf{x}^*) = \sum_{i \in I} \lambda_i \mathbf{u}^i = -\sum_{i \in I} \lambda_i \nabla g_i(\mathbf{x}^*) \qquad (4\text{-}40)$$

其中：I 是刻划起作用约束的集合，并且 $\lambda_i \geqslant 0$ 不全为零。还应该注意到式(4-40)是完全类似于等式约束的拉格朗日乘数方法(4-7)式的。

我们还可以把式(4-40)写成：

$$\nabla F(\mathbf{x}^*) = \sum_{i=1}^{m} \lambda_i \mathbf{u}^i, \quad \lambda_i \geqslant 0 \qquad (4\text{-}41)$$

其中：当 $i \notin I$ 时，$\lambda_i = 0$。另外，由于在 $i \in I$ 时，$g_i(\mathbf{x}^*) = 0$ 而且在 $i \notin I$ 时 $\lambda_i = 0$，所以就还有类似于等式约束极小化问题的结论，即

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) = 0$$

因此，完全和等式约束极小化问题一样，我们在处理线性不等式约束极小化问题时，可以先给出一个包括目标函数和约束函数在内的补助函数为：

$$\Phi(\mathbf{x}) = F(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \qquad (4\text{-}42)$$

则极小点 $\mathbf{x}^*$ 必须满足：

$$\nabla \Phi(\mathbf{x}^*) = \nabla F(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) = 0 \qquad (4\text{-}43)$$

和

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) = 0 \qquad (4\text{-}44)$$

关于线性不等式约束的极小化问题，我们常可以使用线性规划的技巧来联合求解。这里，我们可以提出可行方向法的两种特例方法加以介绍。

1. 梯度投影法

给定 $x\in S$，设包含 x 的最小刻面定义为：

$$g_i(x)=0, \quad i\in I$$

其中：$I=\{i_1, i_2, \cdots, i_p\}$，$p\leqslant m$。如果没有这种约束在 x 起作用，我们可以用无约束最优化方法改进 x，所以我们设定至少有一个约束方程是起作用的。

设 A 的任何一个 $p\times n$ 子矩阵都有满秩 p，$p\leqslant n$。用 A_I 表示 A 的 p 行构成的子矩阵，则 A_I 有 p 秩。因此，$p\times p$ 阵的逆

$$(A_I A_I^T)^{-1}$$

是存在的。

我们现在把通常的正交投影给出最短距离的概念推广到一般空间上去。

对于 R^n 中任何向量 x，在 R^m 中就给出一个子空间 $R(A)$，它是由

$$Ax, \quad x\in R^n$$

的向量构成的。在 R^m 一个向量 y，如果它在 $R(A)$ 上的正交投影是：

$$z=Ax$$

则 $(y-Ax)$ 就是正交于构成 A 的各个列向量的，也就是有：

$$A^T(y-Ax)=0$$

如果 A^TA 有秩 n，则有解：

$$x=(A^TA)^{-1}A^Ty$$

或

$$Ax=A(A^TA)^{-1}A^Ty$$

这就是 $R(A)$ 中的向量。我们称

$$P_A=A(A^TA)^{-1}A^T$$

是把 y 正交投影到 $R(A)$ 的算子。并且，

$$I_m-P$$

就是投影到正交于 A 各个列向量张开的空间的正交子空间上去

的算子。

类似的论证，可以看出秩 p 的一个 $p\times n$ 矩阵 $\mathbf{A}_I$，由它的各个行向量张开的子空间，正交于它的投影算子将是：

$$\mathbf{P}_{A_I}=\mathbf{A}_I^T(\mathbf{A}_I\mathbf{A}_I^T)^{-1}\mathbf{A}_I$$

并且 $\mathbf{I}_n-\mathbf{P}_{A_I}$是正交投影到正交于 $\mathbf{A}_I$ 各行向量张开的子空间。也就是：

$$\mathbf{P}_I=\mathbf{I}_n-\mathbf{A}_I^T(\mathbf{A}_I\mathbf{A}_I^T)^{-1}\mathbf{A}_I \tag{4-45}$$

对 $\mathbf{y}$ 向量作用得：

$$\mathbf{y}-\mathbf{A}_I^T(\mathbf{A}_I\mathbf{A}_I^T)^{-1}\mathbf{A}_I\mathbf{y}=\mathbf{y}-\mathbf{A}_I^T\mathbf{x}$$

它使

$$\mathbf{A}_I(\mathbf{y}-\mathbf{A}_I^T\mathbf{x})=0$$

所以．$\mathbf{P}_I$是把 R^n 中的向量正交投影到由

$$\{\mathbf{x}\mid \mathbf{A}_I\mathbf{x}=0\}$$

的线性子空间上去的算子。而 $\mathbf{A}_I\mathbf{x}=0$ 确定的子空间是和 $g_i(\mathbf{x})=0$，$i\in\mathbf{I}$ 之交所确定的子空间平行的。

梯度投影方法的思想，就是把 $\nabla F(\mathbf{x})$投影到子空间：

（1） $$\Delta\mathbf{x}=\mathbf{P}_I\nabla F(\mathbf{x})$$

其中：$\Delta\mathbf{x}$ 就是 $\nabla F(\mathbf{x})$在包含 $\mathbf{x}$ 的刻面上的投影。

（2） 如果 $\Delta\mathbf{x}\neq 0$，则

$$t_0=\max\{t\mid \mathbf{x}-t\Delta\mathbf{x}\in S\}$$

若 $\mathbf{x}$ 就在 S 的内部，则 $\mathbf{I}$ 是空集，$\mathbf{P}_I=\mathbf{I}$。从而 $\Delta\mathbf{x}=\nabla F(\mathbf{x})$。这种投影方法就是普通的梯度方法。

这样，向量$(\mathbf{x}-t_0\Delta\mathbf{x})$ 就在 S 的边界上，也就是$-t_0\Delta\mathbf{x}$ 给出不离开 S 的在$-\Delta\mathbf{x}$ 方向上的最大移动。定义 t^* 是使

$$\min_{0\leqslant t\leqslant t_0} F(\mathbf{x}-t\Delta\mathbf{x})=f(\mathbf{x}-t^*\Delta\mathbf{x})$$

的实数。并令

$$\mathbf{x}'=\mathbf{x}-t^*\Delta\mathbf{x}$$

若 $t^*<t_0$，起作用的约束方程组并未改变。但是，若 $t^*=t_0$，就达

到S的一个边界，另一个约束方程变为起作用的。如果：

$$g_j(\mathbf{x}')=0,\quad j\notin \mathrm{I}$$

则用包括这些j的改变I，定义出新的$\mathbf{P}_I$，转入（1）。

（3）如果$\Delta\mathbf{x}=0$，则计出：

$$\boldsymbol{\mu}=(\mathbf{A}_I\mathbf{A}_I^T)^{-1}\mathbf{A}_I\nabla F(\mathbf{x})$$

$\nabla F(\mathbf{x})$和$\boldsymbol{\mu}$的关系是：

$$\nabla F(\mathbf{x})=\mathbf{A}_I^T\boldsymbol{\mu}$$

而$\boldsymbol{\mu}$是$\nabla F(\mathbf{x})$到$\mathbf{A}_I$列向量张开的子空间上的正交投影，所以梯度向量可以表示为：

$$\nabla F(\mathbf{x})=\mathbf{A}_I^T\boldsymbol{\mu}=\begin{pmatrix} a_{11} & a_{21} & \cdots & a_{p1}\\ a_{12} & a_{22} & \cdots & a_{p2}\\ \cdots\cdots\cdots & & & \\ a_{1n} & a_{2n} & \cdots & a_{pn}\end{pmatrix}\boldsymbol{\mu}$$

$$=(\nabla g_1(\mathbf{x}),\ \nabla g_2(\mathbf{x}),\cdots,\ \nabla g_p(\mathbf{x}))\boldsymbol{\mu}=\sum_{i=1}^{p}\mu_i\nabla g_i(\mathbf{x})$$

其中

$$\boldsymbol{\mu}^T=(\mu_1,\mu_2,\cdots,\mu_p)$$

或者说，

$$\nabla F(\mathbf{x})+\sum_{i\in \mathrm{I}}(-\mu_i)\nabla g_i(\mathbf{x})=0 \tag{4-46}$$

从式(4-43)看起来，式(4-46)就是以乘数$-\mu_i$($i=1,2,\cdots,p$)为分量的拉格朗日条件，所以当$i\in\mathrm{I}$的$\mu_i\leqslant 0$时，$\mathbf{x}^*$就是最优点。如果$\boldsymbol{\mu}$的某个分量是正的，可以沿着减小$g_i(\mathbf{x})$的方向上移动$\mathbf{x}$以减小$F(\mathbf{x})$，也就是在$g_i(\mathbf{x})<0$的方向上移动$\mathbf{x}$，从而使$g_i(\mathbf{x})=0$变为不起作用，于是从I中把i删去。这样又导出下面两点。

（4）若$\boldsymbol{\mu}$的所有分量是非正的，就停止，这时$\mathbf{x}$是最优。

（5）否则从$\mathbf{A}_I$中去掉$\boldsymbol{\mu}$的最大正分量的那一行，回到（1）中进行。

2. 简化梯度方法

假设在线性不等式约束当中，由系数 a_{ij} 构成的矩阵 $\mathbf{A}$ 是使 $\mathbf{A}$ 的任何一个 n 行子矩阵都具有秩 n。我们引入一组 m 个非负补助变量，正如线性规划中的方法一样，这种变量称为松弛变量，它在不等式方程中起着一种松散的作用使它转化为等式方程，现在则是：

$$\mathbf{A}\mathbf{x}-\mathbf{b}+\mathbf{w}=0$$

这里的 $\mathbf{w}$ 就是 m 维松弛向量。

设 $\mathbf{b}^T=(\mathbf{b}^1, \mathbf{b}^2, \cdots, \mathbf{b}^m)$，$\mathbf{w}^T=(\mathbf{w}^1, \mathbf{w}^2, \cdots, \mathbf{w}^m)$

将 $\mathbf{A}$ 分块为：

$$\left.\begin{aligned}\mathbf{A}_1\mathbf{x}+\mathbf{w}^1=\mathbf{b}^1\\ \mathbf{A}_2\mathbf{x}+\mathbf{w}^2=\mathbf{b}^2\end{aligned}\right\}\quad(4\text{-}47)$$

其中：$\mathbf{A}_1$ 是 $\mathbf{A}$ 的一个 $n\times n$ 子矩阵，$\mathbf{b}^1$ 和 $\mathbf{w}^1$ 是 $\mathbf{b}$ 和 $\mathbf{w}$ 的对应的分块子向量。

一个容许解 $\mathbf{x}$ 是由松弛变量的条件

$$\mathbf{w}^1\geqslant 0,\quad \mathbf{w}^2\geqslant 0\ .$$

来刻画的，松弛变量的零分量就决定了 $\mathbf{x}$ 所在的多面体 S 的刻面。例如，若 $\mathbf{w}$ 的 n 个分量都是零，$\mathbf{x}$ 就是约束多面体 S 的顶点。由假设知道不会有多于 n 个分量都是零的事。

构成式(4-47)的分块方法是使 $\mathbf{w}$ 的零分量都包含在 $\mathbf{w}^1$ 中。在一个刻面上给定 $\mathbf{x}$ 后，若 $\mathbf{w}_j^1=0$ 就在 $\Delta\mathbf{w}_j^1\geqslant 0$ 并且不使 $\mathbf{w}^2$ 向量变为负的方向上保持 $\mathbf{x}$ 的可行性，找出 $F(\mathbf{x})$ 一个局部极小的探索方向。由于 $\mathbf{w}^1$ 给出一种明显的"可行移动"意义，我们就把 $\mathbf{w}^1$ 看成是独立变量，计出 $F(\mathbf{x})$ 对 $\mathbf{w}^1$ 的梯度，也就是：

$$\nabla_{W^1}F(\mathbf{x})^T=-\nabla F(\mathbf{x})^T\mathbf{A}_1^{-1}$$

上式就称为一种"简化梯度"。

所以，一个可行移动可以定义为：

$$\Delta w_j^1=-[\nabla_{W^1}F(\mathbf{x})]_j,\ \text{若}[\nabla_{W^1}F(\mathbf{x})]_j>0\quad\text{或}\quad \mathbf{w}_j^1>0$$

实际上，在任何一个能使 $F(\mathbf{x})$ 减小的方向都可被选用。在

给出一个移动 $\Delta w^1 \geqslant 0$ 以后，式(4-47)就给出：

$$\Delta x = -A_1^{-1}\Delta w^1$$

和

$$\Delta w^2 = A_2 A_1^{-1}\Delta w^1 \tag{4-48}$$

要得到最大的移动长度同时又保持 x 的可行性，w^1 和 w^1 可移动：

$$w^1 + \theta^* \Delta w^1 \quad 和 \quad w^2 + \theta^* \Delta w^2$$

其中

$$\theta^* = \min(\theta_1, \ \theta_2)$$

$$\theta_1 = \max_{\theta \geqslant 0} (\theta | w^1 + \theta w^1 \geqslant 0)$$

$$\theta_2 = \max_{\theta \geqslant 0} (\theta | w^2 + \theta \Delta w^2 \geqslant 0)$$

并保持 x 的可行性。在移动表示式

$$w^1 + \theta_1 \Delta w^1 \qquad w^2 + \theta_2 \Delta w^2 \qquad 0 \leqslant \theta_1, \ \theta_2 \leqslant \theta^*$$

中，我们找出 $F(x)$ 的一个极小：

$$\min_t F(x + t\Delta x)$$

如果完成这种极小化的 t 值是小于 θ^*，这个移动就是完全的，再计一个新的移动方向。否则，选 w^2 的零分量做为 w^1 的分量，把 w^1 的最大分量加以代换，然后对新的独立变量组来计出新的移动方向。

§3 非线性不等式约束问题

我们已经知道，在 $g_i(x)$ 都是线性函数时，任一个从 x^* 进入 S 的向量是由式(4-35)、(4-36)来刻划的，同时，从法卡引理还导出式(4-40)的结果。这个关键性的结论是：

$$\nabla F(x^*) = \sum_{i \in I} \lambda_i u^i = -\sum_{i \in I} \lambda_i \nabla g_i(x^*)$$

但是，在 $g_i(x)$ 是非线性函数时，这个结论不必一定是成立的。

我们先来看下面两个例子：

例 1　极小化

$$F(x_1,\ x_2)=(x_1-2)^2+x_2^2$$

使合于

$$C_1:\quad g_1(x_1,x_2)=-x_1\leqslant 0$$

$$C_2:\quad g_2(x_1,x_2)=-x_2\leqslant 0$$

$$C_3:\quad g_3(x_1,x_2)=-1+x_1^2+x_2\leqslant 0$$

图 4-5 示出问题的可行域 S 和 F 函数的一些等值曲线。显然，

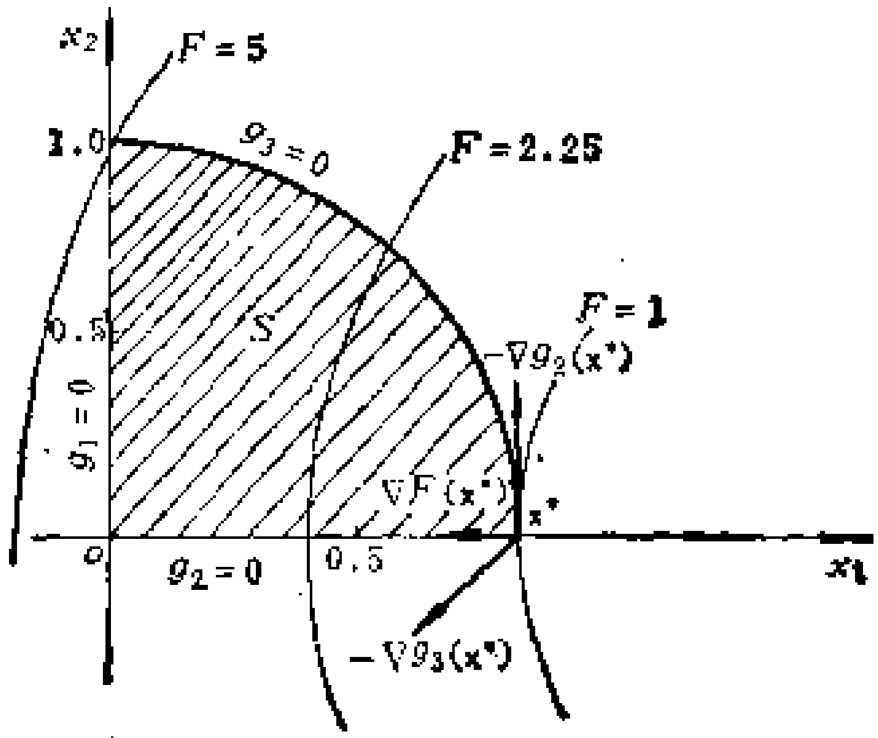


图4-5　例 1 存在拉格朗日乘数图示

极小点是在 $x^*=(1,0)^T$，起作用的约束是 C_2 和 C_3，也就是$I=\{2,3\}$。在极小点有：

$$\nabla F(x^*)=(-4,\ 0)^T$$

$$-\nabla g_2(x^*)=(0,\ 1)^T$$

$$-\nabla g_3(x^*)=(-2,\ -1)^T$$

而这些向量的方向表在图 4-5 中，在取

$$\lambda_1=0,\quad \lambda_2=\lambda_3=2$$

时，我们就有：

$$\nabla F(x^*)+\lambda_1\nabla g_1(x^*)+\lambda_2\nabla g_2(x^*)+\lambda_3\nabla g_3(x^*)=0$$

由于 C_2 和 C_3 是在 x^* 起作用的，所以有：

$$g_2(\mathbf{x}^*)=g_3(\mathbf{x}^*)=0$$

但是 $$g_1(\mathbf{x}^*)<0$$

所以这些 λ 值是满足下式：

$$\sum_{i=1}^{3}\lambda_i g_i(\mathbf{x}^*)=0$$

例 2　根据 1951 年库恩-特克(Kuhn-Tucker)给出的例题，

极小化　　$F(x_1, x_2)=(x_1-2)^2+x_2^2$

使合于

$$C_1:\ g_1(x_1, x_2)=-x_1\leqslant 0$$

$$C_2:\ g_2(x_1, x_2)=-x_2\leqslant 0$$

$$C_3:\ g_3(x_1, x_2)=x_2-(1-x_1)^3\leqslant 0$$

图 4-6 表出这个问题的可行域 S 和 F 的一些等值曲线，这个问题的答案是：

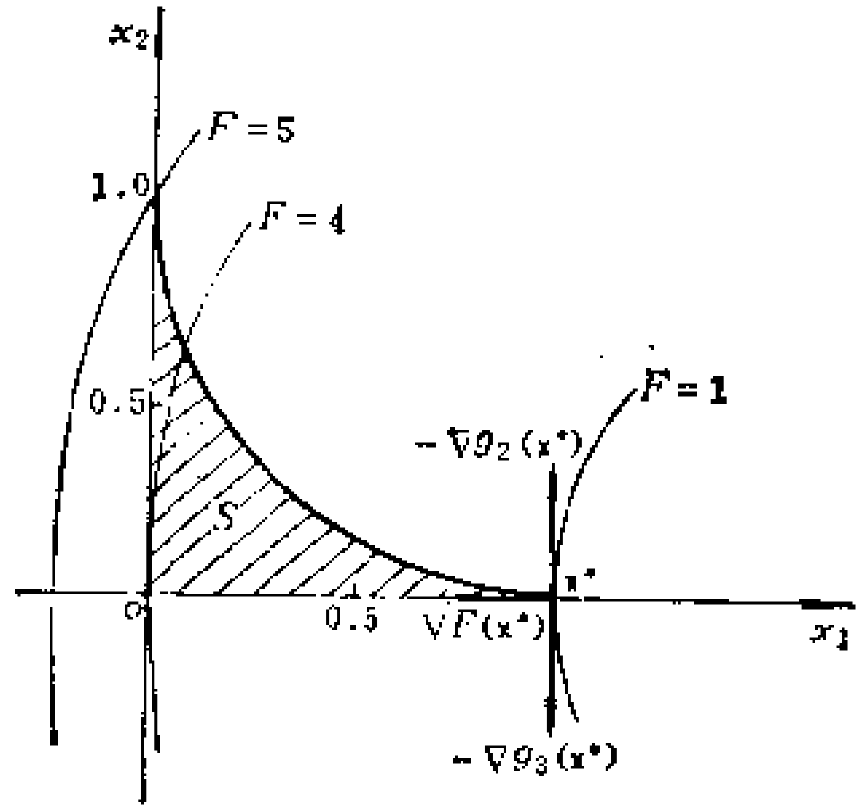


图4-6　例 2 没有拉格朗日乘数图示

$$\mathbf{x}^*=(1,0)^T,\ \nabla F(\mathbf{x}^*)=(-2,0)^T$$

在 $\mathbf{x}^*$ 上，约束 C_2 和 C_3 是起作用的，也就是 $I=\{2,3\}$，并且

$$\nabla g_2(\mathbf{x}^*)=(0,-1)^T,\ \nabla g_3(\mathbf{x}^*)=(0,1)^T$$

这两个起作用约束函数的梯度向量是线性相关的，所以不存在 λ_2

和 λ_3 能使下式成立：

$$\nabla F(\mathbf{x}^*)+\lambda_2\nabla g_2(\mathbf{x}^*)+\lambda_3\nabla g_3(\mathbf{x}^*)=0$$

因为 ∇F 有一个非零的第一分量，而 ∇g_2 和 ∇g_3 的第一分量都是零。如果用 $\lambda=0$ 乘上 $\nabla F(\mathbf{x}^*)$，则对任何 $\lambda_2=\lambda_3$ 就恒有：

$$\lambda\nabla F(\mathbf{x}^*)+\lambda_2\nabla g_2(\mathbf{x}^*)+\lambda_3\nabla g_3(\mathbf{x}^*)=0$$

我们在第一节中早已指出等式约束的梯度向量只有在它们是线性独立时，才能解出 $\lambda\neq0$ 的 λ_2，λ_3 解。现在 $\nabla g_2(\mathbf{x}^*)$，$\nabla g_3(\mathbf{x}^*)$ 却是线性相关的。由此可见，使拉格朗日乘数存在的必要条件是约束函数要满足一定的“约束合格条件”，约束函数的梯度在 $\mathbf{x}^*$ 是线性独立这个条件就是一种约束合格条件。

其他的约束合格条件可以由下面一些定理给出，它们的证明放在本节的最后。

1．弗里兹-约翰(Fritz-John)微分形式定理

设 $F(\mathbf{x}), g_i(\mathbf{x})(i=1,2,\cdots,m)$ 在 R^n 中包含 $\mathbf{x}^*$ 的某一开集内有连续的偏导数，若 $\mathbf{x}^*$ 是 $F(\mathbf{x})$ 合于 $g_i(\mathbf{x})\leqslant0(i=1,2,\cdots,m)$ 的一个带约束条件的极小点，则有不全为零的 λ_i $(i=0,1,2,\cdots,m)$ 满足下式：

$$\left.\begin{aligned}\lambda_0\nabla F(\mathbf{x}^*)+\sum_{i=1}^{m}\lambda_i\nabla g_i(\mathbf{x}^*)=0\\ \sum_{i=1}^{m}\lambda_i g_i(\mathbf{x})=0\\ \lambda_i\geqslant0\qquad i=0,1,2,\cdots,m\end{aligned}\right\}\tag{4-49}$$

注意，这里的 λ_0 可能是零。如果 λ_0 是零，这个定理就失去了我们所感兴趣的有关带约束极小化问题的联系。所以我们还要给出保证 $\lambda_0\neq0$ 的一种约束合格条件的形式。

对于起作用约束附标集合 I，$\nabla g_i(\mathbf{x}^*)$，$i\in I$ 是线性独立向量就是一种线性独立的约束合格条件的假定。在这种线性独立假定下，我们就可以有 $\lambda_0\neq0$，因为，否则 λ_1，λ_2，$\cdots$，λ_m 将全

部是零。这时就可以令 $\lambda_0=1$，或者说在式(4-49)的结论中用λ_0遍除各项，并将λ_i/λ_0仍用λ_i标出，就得到弗里兹-约翰定理的一种特殊情形。

2. 库恩-特克(Kuhn-Tuker)微分形式定理

设$F(\mathbf{x})$，$g_i(\mathbf{x})$（$i=1, 2, \cdots, m$）在R^n中包含$\mathbf{x}^*$的某一开集内有连续偏导数。若$\mathbf{x}^*$是满足线性独立约束合格条件的一些约束$g_i(\mathbf{x})\leqslant 0$（$i=1, 2, \cdots, m$），极小化了$F(\mathbf{x})$的点，则有非负的拉格朗日乘数$\lambda_1$，$\lambda_2$，$\cdots$，$\lambda_m$使下式成立：

$$\nabla F(\mathbf{x}^*)+\sum_{i=1}^{m}\lambda_i\nabla g_i(\mathbf{x}^*)=0$$

$$\sum_{i=1}^{m}\lambda_i g_i(\mathbf{x}^*)=0$$

$$\lambda_i\geqslant 0 \qquad i=1, 2, \cdots, m$$

现在，定义拉格朗日函数为：

$$\Phi(\mathbf{x}, \boldsymbol{\lambda})=F(\mathbf{x})+(\boldsymbol{\lambda}, g(\mathbf{x}))$$

其中

$$\boldsymbol{\lambda}=(\lambda_1, \lambda_2, \cdots, \lambda_m)^T$$

$$g(\mathbf{x})=(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_m(\mathbf{x}))^T$$

那么，库恩-特克定理就可以清楚的表达为：

$$\nabla_x\Phi(\mathbf{x}^*, \boldsymbol{\lambda})=0$$

$$(\boldsymbol{\lambda}, \nabla_\lambda\Phi(\mathbf{x}^*, \boldsymbol{\lambda}))=0$$

其中：∇_x表示对于$\mathbf{x}$的梯度，∇_λ表示对于$\boldsymbol{\lambda}$的梯度。

库恩-特克定理是对应于拉格朗日乘数定理（4-7）的。在式（4-7）中的拉格朗日乘数是不限定符号的，但在库恩-特克定理中则是限定为非负的。这种区别产生的缘因是，我们可以把一个等式约束$h(\mathbf{x})=0$写成两个不等式约束$h(\mathbf{x})\leqslant 0$和$-h(\mathbf{x})\leqslant 0$，从而拉格朗日函数将是：

$$\Phi(\mathbf{x})=F(\mathbf{x})+(\lambda^+, h(\mathbf{x}))+(\lambda^-, -h(\mathbf{x}))$$

其中：$\lambda^+\geqslant 0$，$\lambda^-\geqslant 0$，或者说：

$$\Phi(x)=F(x)+(\lambda,\ h(x))$$

其中：$\lambda=\lambda^+-\lambda^-$，所以就不必限定$\lambda$的符号。

实际上，我们所得到的拉格朗日乘数$(\lambda_0,\ \lambda)\triangleq(\lambda_0,\ \lambda_1,\ \lambda_2,\ \cdots,\ \lambda_m)$表示了分离的超平面的法线向量。所以，拉格朗日乘数不必是唯一的，也就是可以有多于一个的分离超平面。因此，一般的还可以再附加一些设定。

早在1951年库恩-特克最先提出约束合格条件的概念，他们引用的不是我们在上面提出的线性独立约束合格条件，他们提出的约束合格条件包括一些“进入向量”和“可微弧段”的概念，是不容易验证的。我们在下面将另引用司勒特（Slater）约束合格条件，这种条件是容易验证的。这种条件在S是凸的而且$F(x)$和$g_1(x)$，$g_2(x)$，$\cdots$，$g_m(x)$都是凸函数时就可以应用。

关于集合的是凸的定义，我们是已经在上面讨论过的。

一个函数$F(x)$在一个凸集S上是凸的意思是指不等式

$$F(\mu x^1+(1-\mu)x^2)\leqslant\mu F(x^1)+(1-\mu)f(x^2) \qquad (4\text{-}50)$$

对于所有x^1，$x^2\in S$，$0\leqslant\mu\leqslant 1$成立，或者说

$$F(x^1+\mu h)\leqslant\mu F(x^1+h)+(1-\mu)F(x^1) \qquad (4\text{-}51)$$

对于所有x^1，$x^1+h\in S$，$0\leqslant\mu\leqslant 1$成立。

由于S是一个凸集，所以当x^1，$x^2\in S$时，$\mu x^1+(1-\mu)x^2$在$0\leqslant\mu\leqslant 1$时也将在凸集S中，所以式（4-50）的左边是有定义的。

从一元函数来看，凸函数和极小点存在的关系是很清楚的，几何意义是：由式(4-50）的关系说明该点在函数的图示上任何二点之间，函数图形落在联接$(x^1,\ F(x^1))$和$(x^2,\ F(x^2))$二点所得弦的下面，图4-7表示x是纯量变量的情况，P点的坐标是：

$$[\mu x^1+(1-\mu)x^2,\ \mu F(x^1)+(1-\mu)F(x^2)]$$

设S是在R^n中的一个凸子集，如果凸函数$g(x)$有在S中

的 x 使 $g(x)<0$，则称 $g(x)$ 在 S 上满足司勒特约束合格条件。

在 $F(x)$ 和 $g_i(x)(i=1, 2, \cdots, m)$ 不满足弗里兹-约

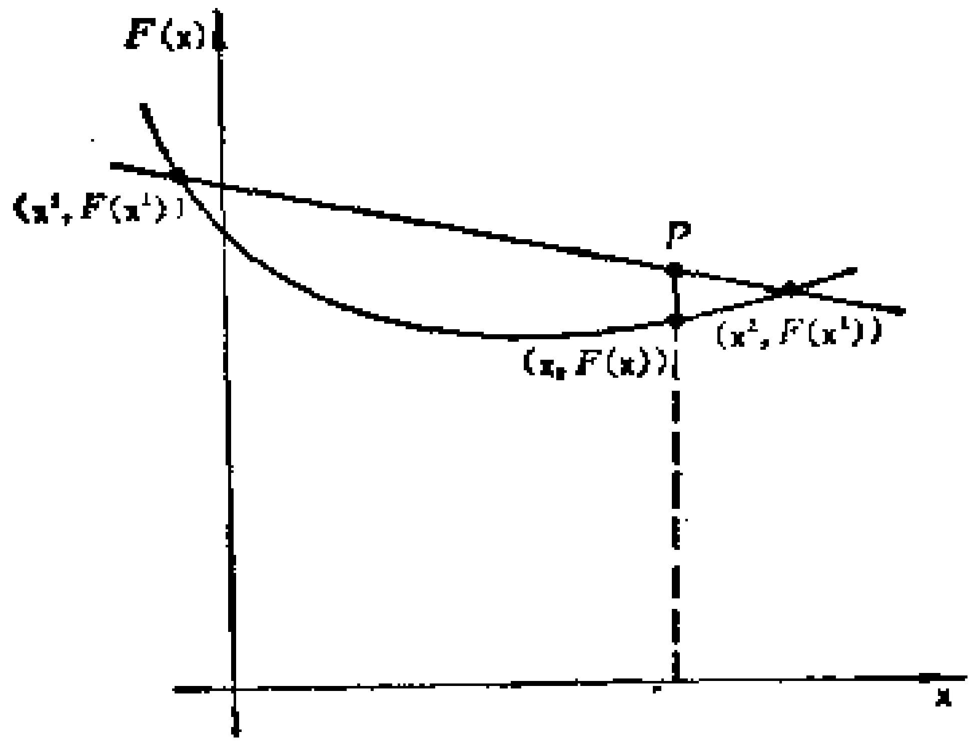


图4-7　一个凸函数的图示

翰定理和库恩-特克定理的连续可微性的要求时，我们可以代之以可行域 S 是凸集和 $F(x)$，$g_1(x)$，$g_2(x)$，$\cdots$，$g_m(x)$ 都是在 S 上的凸函数以构成鞍点定理。现在，我们讨论的带约束的极小化问题是：

$$\min F(x),\quad x\in R^n$$

使合于

$$g_i(x)\leqslant 0 \qquad i=1, 2, \cdots, m \tag{4-52}$$

其中：$F(x)$，$g_i(x)(i=1, 2, \cdots, m)$ 是在凸集 S 上的凸函数。

若定义拉格朗日函数为：

$$\Phi(x, \lambda_0, \lambda)=\lambda_0 F(x)+(\lambda, g(x)) \tag{4-53}$$

3．弗里兹-约翰鞍点定理

由式（4-53），又设 x^* 是有约束极小化问题（4-52）的一解，则有 $\lambda_0^*\geqslant 0$ 和 $\lambda^*\geqslant 0$ 使

$$\Phi(\mathrm{x}^*,\ \lambda_0^*,\ \lambda) \leqslant \Phi(\mathrm{x}^*,\ \lambda_0^*,\ \lambda^*) \leqslant \Phi(\mathrm{x},\ \lambda_0^*,\ \lambda^*) \qquad (4\text{-}54)$$

对所有 $\lambda \geqslant 0$ 和 S 中的所有 x 成立，并且：

$$(\lambda^*,\ g(\mathrm{x}^*)) = 0 \qquad (4\text{-}55)$$

正如在弗里兹-约翰定理的微分形式中所指出的那样，我们还要指出在附加一种适当的约束合格条件以后，就能有 $\lambda_0^* > 0$，然后令 $\lambda_0^* = 1$ 就得到下面定理。

4．库恩-特克鞍点定理

若 x^* 是约束函数满足司勒特约束合格条件的带约束极小化问题的一解，则有 $\lambda^* \geqslant 0$ 使

$$\Psi(\mathrm{x}^*,\ \lambda) \leqslant \Psi(\mathrm{x}^*,\ \lambda^*) \leqslant \Psi(\mathrm{x},\ \lambda^*) \qquad (4\text{-}56)$$

对所有 $\lambda \geqslant 0$ 以及所有 S 中的 x 成立，并且

$$(\lambda^*,\ g(\mathrm{x}^*)) = 0 \qquad (4\text{-}57)$$

其中

$$\Psi(\mathrm{x},\ \lambda) = F(\mathrm{x}) + (\lambda,\ g(\mathrm{x})) \qquad (4\text{-}58)$$

弗里兹-约翰微分形式定理的证明

因为对于充分小的 $t > 0$，则有：

$$F(\mathrm{x}^* + t\mathrm{z}) = F(\mathrm{x}^*) + t\ (\nabla F(\mathrm{x}^*), \mathrm{z}) + o(t)$$

其中：$o(t)$ 表示当 $t \to 0$ 时，$o(t)/t \to 0$ 的一个无穷小。设 I 是起作用的约束附标集合，则：

$$g_i(\mathrm{x}^* + t\mathrm{z}) = g_i(\mathrm{x}^*) + t\ (\nabla g_i(\mathrm{x}^*), \mathrm{z}) + o(t)$$
$$= t\ (\nabla g_i(\mathrm{x}^*), \mathrm{z}) + o(t), \qquad i \in \mathrm{I}$$

因此，不等式

$$(\nabla F(\mathrm{x}^*),\ \mathrm{z}) < 0$$
$$(\nabla g_i(\mathrm{x}^*), \mathrm{z}) < 0, \quad i \in \mathrm{I}$$

是互不相容的，因为，若对充分小的 $t > 0$，就使

$$F(\mathrm{x}^* + t_z) < F(\mathrm{x}^*) \qquad (4\text{-}59)$$
$$g_i(\mathrm{x}^* + t_z) \leqslant 0 \qquad i = 1, 2, \cdots, m$$

对某 z 成立，这是和设定 x^* 为最优点相矛盾的。

实际上，我们可以把式（4-59）的约束方程分为两类，要求证实

$$(\nabla F(\mathrm{x}^*), \mathrm{z}) < 0 \qquad (4\text{-}60)$$

$$(\nabla g_k(x^*), z) \leqslant 0, \quad k \in I_1 \tag{4-61}$$

$$(\nabla g_j(x^*), z) < 0, \quad j \in I - I_1 \tag{4-62}$$

无解，其中 I_1 定义为：

$$I_1 = \{k | g_k(x^*) = 0 \text{ 并且 } g_k \text{ 在 } x^* \text{ 是凹的}\}$$

所以，g_k（•）是凹的意思，$-g_k$（•）是凸的意思。关于非凹 约束方程，用泰勒展开，由定理的假设以及式（4-62）可知，对于充分小的 t 能有：

$$g_j(x^* + tz) < 0, \quad j \in I - I_1$$

对于凹约束方程，从凸函数的性质有：

$$f(x+h) \geqslant f(x) + (\nabla f(x), h)$$

并就式（4-61）用严格的不等式代入，就有：

$$g_k(x^* + tz) \leqslant 0, \quad k \in I_1$$

不起作用的约束对充分小的 t 仍在 $x^* + tz$ 不起作用，从而 $x^* + tz$ 对充分小的 t 是可行的，但是 $F(x^* + tz) < F(x^*)$ 就与 x^* 是一个局部极小的假设相矛盾。

从法卡引理容易知道，对于任一个矩阵 $A \neq 0$ 和 B，或者是

$$\begin{aligned} Ax &< 0 \\ Bx &\leqslant 0 \end{aligned} \tag{4-63}$$

有一解，或者

$$\begin{gathered} \lambda^T A + \mu^T B = 0 \\ \lambda \geqslant 0 \qquad \lambda \neq 0 \\ \mu \geqslant 0 \end{gathered} \tag{4-64}$$

有一解，但决不是两者同时成立。

要证明弗里兹-约翰定理，只需要用这样的论断：若式（4-63）是互不相容的，则式（4-64）就有一个解。

以行向量 $\nabla F(x^*)^T$ 和 $\nabla g_k(x^*)^T$，$k \in I_1$ 构成 A，而以行向量 $\nabla g_j(x^*)^T$，$j \in I - I_1$ 构成 B，就知道有 λ_0 和 λ 使下式成立：

$$\lambda_0 \nabla F(x^*) + \sum_{i \in I_1} \lambda_i \nabla g_k(x^*) + \sum_{i \in I - I_1} \lambda_i \nabla g_i(x^*) = 0$$

其中

$$\binom{\lambda_0}{\lambda_{I-I_1}} \geqslant 0 \quad \text{而不恒等于零，且} \lambda_{I_1} \geqslant 0$$

然后，对 $i \notin I$ 定义 $\lambda_i = 0$，就证得此定理。

弗里兹-约翰鞍点定理的证明

我们完全可以按照§1中从可行方向法导出拉格朗日乘数法则那样，应用樊鰀的理论来进行论证。

首先，注意到

$$F(x) - F(x^*) < 0$$
$$g(x) \leqslant 0$$

在S中无解，因为x^*是在S中的有约束极小点。还注意到这种结论等于说

$$F(x) - F(x^*) < 0$$
$$g(x) < 0$$

在S中无解。

现在，对每一个$x \in S$定义一个集合V_x是：

$$V_x = \left\{ \begin{pmatrix} v_0 \\ v \end{pmatrix};\ v_0 \in R,\ v \in R^m,\ F(x) - F(x^*) < v_0,\ g(x) < v \right\}$$

并定义

$$V = \bigcup_{x \in S} V_x ●$$

由假设可知V不包含原点。集合V是凸的，因为：

$$\begin{pmatrix} v_0^1 \\ v^1 \end{pmatrix} \in V_{x^1} \text{ 和 } \begin{pmatrix} v_0^2 \\ v^2 \end{pmatrix} \in V_{x^2} \text{ 就有} \begin{pmatrix} v_0^\theta \\ v^\theta \end{pmatrix} \in V_{x^\theta}$$

其中

$$v_0^\theta = \theta v_0^1 + (1-\theta) v_0^2$$
$$v^\theta = \theta v^1 + (1-\theta) v^2 \qquad 0 < \theta < 1$$
$$x^\theta = \theta x^1 + (1-\theta) x^2$$

这时，就有关于V的一个分离超平面对所有的$\begin{pmatrix} v_0 \\ v \end{pmatrix} \in V$，$\begin{pmatrix} \lambda_0^* \\ \lambda^* \end{pmatrix} \neq 0$有：

$$\lambda_0^* v_0 + (\lambda^*, v) \geqslant 0 \tag{4-65}$$

由于v_0和v的每一个分量可取得很大，所以必须有$\lambda_0^* \geqslant 0$和$\lambda^* \geqslant 0$。

就任意一个ε ($\varepsilon > 0$)，取

$$v_0 = F(x) - F(x^*) + \varepsilon$$
$$v = g(x) + \boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon} = \varepsilon(1, 1, \cdots, 1)^T$$

● 所有$x \in S$的V_x的并集，写成$\bigcup_{x \in S} V_x$。

则式（4-65）就给出：

$$\lambda_0^*[F(x)-F(x^*)]+\lambda_0^*\bar{\varepsilon}+(\lambda^*,\ g(x))+\bar{\varepsilon}\sum_{i=1}^{m}\lambda_i^*\geqslant 0$$

但是 $\bar{\varepsilon}$ 是任意的，所以有：

$$\lambda_0^*[F(x)-F(x^*)]+(\lambda^*,g(x))\geqslant 0 \tag{4-66}$$

从 $\lambda^*\geqslant 0$ 和 $g(x)\leqslant 0$ 得知，对于 S 中所有的 x 有 $(\lambda^*,\ g(x))\leqslant 0$，特别是 $(\lambda^*,\ g(x^*))\leqslant 0$，取 x 等于 x^*，从式（4-66）就得到 $(\lambda^*,\ g(x^*))\geqslant 0$。所以：

$$(\lambda^*,\ g(x^*))=0 \tag{4-67}$$

这样，从式（4-66）和（4-67），我们就得到定理右边的不等式关系，即

$$\lambda_0^*F(x^*)+(\lambda^*,g(x^*))\leqslant\lambda_0^*F(x)+(\lambda^*,g(x))$$

再因为 $g(x^*)\leqslant 0$，对于任一 $\lambda\geqslant 0$ 就有 $(\lambda,\ g(x^*))\leqslant 0$，所以就得到定理的左边不等式关系，即

$$\lambda_0^*F(x^*)+(\lambda,\ g(x^*))\leqslant\lambda_0^*F(x^*)=\lambda_0^*F(x^*)+(\lambda^*,g(x^*))$$

这里，要保证 $\lambda_0^*>0$，我们还要设有司勒特约束合格条件。

事实上，假设 $\lambda_0^*=0$，则 $\lambda^*\geqslant 0$，并且弗里兹-约翰鞍点定理变为对 S 中所有 x 有：

$$0=(\lambda^*,\ g(x^*))\leqslant(\lambda^*,\ g(x)) \tag{4-68}$$

但是司勒特约束合格条件就说明有使 $g(x)<0$ 或 $(\lambda^*,\ g(x))<0$ 的 x 存在，它与式（4-68）矛盾，所以弗里兹-约翰定理的假设加上司勒特约束合格条件就证明了库恩-特克鞍点定理。

容易看出，在 $\lambda^*\geqslant 0$，$x^*\in S$ 的 $(\lambda^*,\ x^*)$ 满足库恩-特克定理时，x^* 就解决了约束极小化问题。事实上，从不等式的左边，我们知道对所有的 $\lambda\geqslant 0$ 有：

$$(\lambda-\lambda^*,\ g(x^*))\leqslant 0,\ \lambda\geqslant 0 \tag{4-69}$$

所以就有 $g(x^*)\leqslant 0$，加之 $\lambda^*\geqslant 0$，故有 $(\lambda^*,\ g(x^*))\leqslant 0$。在式(4-69)中令 $\lambda=0$，我们就得到 $(\lambda^*,\ g(x^*))\geqslant 0$，因此

$$(\lambda^*,\ g(x^*))=0$$

那么，不等式的右边就可化为：

$$F(x^*)\leqslant F(x)+(\lambda^*,g(x))$$

但是 $(\lambda^*,\ g(x))\leqslant 0$，故对一切使 $g(x)\leqslant 0$ 的 $x\in S$ 有 $F(x^*)\leqslant F(x)$。所以，x^* 是有约束的极小点。所有上述有关充分性的论证都没有用到凸性。

§4 序列无约束极小化方法

前述三节都是针对用拉格朗日乘数把原来的有约束极小化问题转化为拉格朗日函数的求解，探讨它们求解的条件。对于等式约束的极小化问题，从理论上说，在目标函数和约束函数满足一定条件时，是能够采用拉格朗日函数求出带约束的极小点的；关于不等式约束的极小化问题，在使用拉格朗日函数时，还要对约束函数有些合格条件的限制。但是，使用拉格朗日函数的计算问题至今还未能有明确的方法，使用起来很不方便。

序列无约束极小化方法就是把通常带约束的极小化问题转化为一序列的无约束极小化问题，不是按拉格朗日乘数方法一次同时求出乘数和极小点，而是每次给定一些乘数，求出极小点，然后限制乘数的一序列选法，找到对应极小点序列所收敛的极小点，就得到带约束的极小点的解。

典型的有约束极小化问题是：

极小化 $F(\mathbf{x})$

$$\text{使合于}\quad h_i(\mathbf{x})=0 \qquad i=1,\ 2,\ \cdots,\ p \tag{4-70}$$
$$g_j(\mathbf{x})\geqslant 0 \qquad j=1,\ 2,\ \cdots,\ m$$

其中：$F(\mathbf{x})$，$g_1(\mathbf{x})$，$g_2(\mathbf{x})$，…，$g_m(\mathbf{x})$及$h_1(\mathbf{x})$，$h_2(\mathbf{x})$，…，$h_p(\mathbf{x})$ 都是 R^n 中向量 $\mathbf{x}$ 的实值函数。

关于序列无约束极小化方向的外点法和内点法的收敛准则，我们可以由下列三个定理给出。

1．外点法

（1）序列极小化之间的关系

针对式（4-70）的有约束极小化问题，我们选用惩罚函数是：

$$P(\mathbf{x},\ \rho_k)=F(\mathbf{x})+r(\rho_k)S(\mathbf{x}) \tag{4-71}$$

其中：$\rho_k\geqslant\rho_{k+1}>0$，且 r（$r>0$）随 ρ 的减少而增加，序列 $r(\rho_k)$ 当 $k\to\infty$ 时，$r(\rho_k)\to\infty$，并且：

$$S(\mathbf{x})=\sum_{i=1}^{m}|\min[0,\ g_i(\mathbf{x})]|^{\alpha}+\sum_{j=1}^{p}|h_j(\mathbf{x})|^{\beta}$$

$$\alpha\geqslant 1,\quad \beta\geqslant 1$$

可以证明：若$P(\mathbf{x},\ \rho_k)$和$P(\mathbf{x},\ \rho_{k+1})$在$R^n$中各在$\mathbf{x}^{k*}$和$\mathbf{x}^{k+1*}$达极小，则有：

$$\begin{aligned}P(\mathbf{x}^{k+1*},\ \rho_{k+1})&\geqslant P(\mathbf{x}^{k*},\ \rho_k)\\ S(\mathbf{x}^{k*})&\geqslant S(\mathbf{x}^{k+1*})\\ F(\mathbf{x}^{k+1*})&\geqslant F(\mathbf{x}^{k*})\end{aligned}\tag{4-72}$$

证明：

因为$r(\rho)S(\mathbf{x})\geqslant 0$，故有：

$$\begin{aligned}P(\mathbf{x}^{k+1*},\ \rho_{k+1})&=F(\mathbf{x}^{k+1*})+r(\rho_{k+1})S(\mathbf{x}^{k+1*})\\ &\geqslant F(\mathbf{x}^{k+1*})+r(\rho_k)S(\mathbf{x}^{k+1*})\end{aligned}$$

但是$\mathbf{x}^{k*}$是极小化了$F(\mathbf{x},\ \rho_k)$的，所以有：

$$\begin{aligned}&F(\mathbf{x}^{k+1*})+r(\rho_k)S(\mathbf{x}^{k+1*})\\ &\geqslant F(\mathbf{x}^{k*})+r(\rho_k)S(\mathbf{x}^{k*})\end{aligned}\tag{4-73}$$

因此就有式（4-72）中的第一个结论，即

$$P(\mathbf{x}^{k+1*},\ \rho_{k+1})\geqslant P(\mathbf{x}^{k*},\ \rho_k)$$

其次，从$\mathbf{x}^{k+1*}$的意义有：

$$F(\mathbf{x}^{k+1*})+r(\rho_{k+1})S(\mathbf{x}^{k+1*})\leqslant F(\mathbf{x}^{k*})+r(\rho_{k+1})S(\mathbf{x}^{k*})\tag{4-74}$$

将式（4-73），（4-74）相加，就有：

$$r(\rho_k)[S(\mathbf{x}^{k*})-S(\mathbf{x}^{k+1*})]\leqslant r(\rho_{k+1})[S(\mathbf{x}^{k*})-S(\mathbf{x}^{k+1*})]$$

但是$r(\rho_k)\leqslant r(\rho_{k+1})$，这样就有式（4-72）的第二个结论，即

$$S(\mathbf{x}^{k*})\geqslant S(\mathbf{x}^{k+1*})$$

最后，从（4-73）式给出：

$$F(\mathbf{x}^{k+1*})-F(\mathbf{x}^{k*})\geqslant r(\rho_k)[S(\mathbf{x}^{k*})-S(\mathbf{x}^{k+1*})]$$

而$r(\rho_k)>0$，$S(\mathbf{x}^{k*})\geqslant S(\mathbf{x}^{k+1*})$，所以知道：

$$F(x^{k+1*})-F(x^{k*})\geqslant 0$$

因此，可得到式（4-72）的第三个结论，即

$$F(x^{k+1*})\geqslant F(x^{k*})$$ 证毕。

（2）序列无约束极小点的收敛性

设式（4-70）中的可行集X非空，且有$\varepsilon>0$，使集合

$$X^{\varepsilon}=\{x|x\in R^n,\ g_i(x)\geqslant-\varepsilon\quad i=1,2,\cdots,m,\ |h_j(x)|\leqslant\varepsilon\quad j=1,2,\cdots,p\}$$

是紧的，也就是在X^{ε}中任何一个收敛子序列都将收敛于X^{ε}中一元；设$P(x,\rho_k)$对一切k在R^n中能找到它的无约束极小x^{k*}；若$\{\rho_k\}$是收敛于零的正数的严格下降序列，则有序列最优解的一个收敛子序列$\{x^{k_i*}\}$存在，并且任何一个这样的收敛子序列的极限就是问题的最优解。

证明：

由式（4-63）可知，序列$P(x^{k*},\rho_k)$是一个升序列，并且因为X是紧的，所以$F(x)$是连续的，也就是至少有一点$x^*\in X$使$F(x)$达到带约束的极小，这时$S(x^*)=0$，所以有：

$$F(x^*)=F(x^*)+r(\rho_k)S(x^*)\geqslant P(x^{k*},\rho_k)$$

而序列$\{P(x^{k*},\rho_k)\}$是界于上的，故收敛于P^0，类似地，序列$\{F(x^{k*})\}$是递增的，并且：

$$F(x^{k*})\leqslant F(x^{k*})+r(\rho_k)S(x^{k*})=F(x^{k*},\rho_k)$$

因此，

$$F(x^{k*})\leqslant F(x^*)$$

所以，$\{F(x^{k*})\}$收敛于极限F^0。但是，由于：

$$\lim_{k\to\infty}\{r(\rho_k)S(x^{k*})\}=F^0-f^0$$

这就必然是：

$$\lim_{k\to\infty}\{S(x^{k*})\}=0$$

于是，对每一个$\delta>0$，有一个自然数$K(\delta)$，使当$k\geqslant K(\delta)$时$x^{k*}\in X^{\delta}$，从而对于充分大的$\hat{K}(\varepsilon)$，点x^{k*}对一切$k\geqslant\hat{K}(\varepsilon)$

将在一个紧集 X' 中，所以有一个子序列$\{x^{k_i*}\}$收敛于极限 x^0，且将$S(x^0)=0$，因此 $x^0\in X$。

从 x^*的最优性得知：

$$F(x^*)\leqslant F(x^0)$$

也就是，对一切 k_i 有：

$$F(x^{k_i*})\leqslant F(x^{k_i*})+r(\rho_{k_i})S(x^{k_i*})\leqslant F(x^*)$$
$$+r(\rho_{k_i})S(x^*)=F(x^*)$$

所以

$$\lim_{k\to\infty}\{F(x^{k_i*})\}=F(x^0)\leqslant F(x^*)$$

故

$$F(x^0)=F(x^*)$$

因此，x^0 就是问题的解。证毕。

2．内点法

设：仅就不等式约束，所以构成的惩罚函数为：

$$P(x,\ \rho_k)=F(x)+t(\rho_k)q(x) \tag{4-75}$$

其中：ρ_k 是趋于零而大于零的序列，并且：

$$\lim_{k\to\infty}t(\rho_k)=0,\qquad \lim_{k\to\infty}\{q(x^k)\}=\infty$$

例如，取

$$t_1(\rho)=\rho \quad 或 \quad t_2(\rho)=(\rho)^2$$

取

$$q_1(x)=-\sum_{i=1}^{m}\lg g_i(x)$$

或

$$q_2(x)=\sum_{i=1}^{m}\frac{1}{g_i(x)},\qquad q_3(x)=\sum_{i=1}^{m}\frac{1}{[g_i(x)]^2}$$

等等。

如果可行域$X\subset R^n$是闭的，它的内点集合 X^0 非空，X是

X^0的闭包；有$X^0 \in X$具$F(x^0)=\alpha^0$，使集合$S(F,\ \alpha^0)\cap X$是紧的●，其中$S(F,\ \alpha^0)$是F在α^0的水平集合。

若q函数对$x \in X^0$是正的，且若$P(x,\ \rho_k)$对一切k在X^0中能达到无约束的极小，并且$\{\rho_k\}$是收敛于零的正数严格下降序列，则有序列最优解的一收敛子序列$\{x^{k_j *}\}$，且任一该序列的极限就是问题的最优解。

证明：

从所给的条件可知，$S(F,\ \alpha^0)\cap X$非空且是紧的，所以连续函数F在$x^* \in X$对x达极小值$F(x^*)$。设$P(x^{k*},\ \rho_k)$是极小，则

$$P(x^{0*},\ \rho_0) \geqslant P(x^{1*},\ \rho_1) \geqslant \cdots\cdots \geqslant F(x^*)$$

而$\{P(x^{k*},\ \rho_k)\}$是界于下的严格下降序列，它收敛于极限$\hat{p} \geqslant F(x^*)$。

设$\hat{p} > F(x^*)$，从所给条件以及F的连续性，就应有$\delta > 0$以及球邻域$N_\delta(x^*)$使$X^0 \cap N_\delta(x^*) \neq \Phi$且

$$F(x) \leqslant \hat{p} - \frac{1}{2}[\hat{p} - F(x^*)]$$

对一切$x \in N_\delta(x^*)$成立。今取任一点$\bar{x} \in X^0 \cap N_\delta(x^*)$，则有自然数$K$，对每一$k \geqslant K$有：

$$t(\rho_k)\, q(\bar{x}) < \frac{1}{4}[\hat{p} - F(x^*)]$$

故

$$P(x^{k*},\ \rho_k) \leqslant F(\bar{x}) + t(\rho_k)\, q(\bar{x}) < \hat{p} - \frac{1}{4}[\hat{p} - F(x^*)]$$

对一切$k \geqslant K$成立。那么，它和$\{P(x^{k*}, \rho_k)\}$的单调收敛于$\hat{p}$的性质相矛盾，故

$$\hat{p} = F(x^*)$$

另外，从条件可知，有$\hat{K}$对一切$k \geqslant \hat{K}$，点x^{k*}是在一个紧

● $S(F, \alpha^0) \cap X$表示集合$S(F, \alpha^0)$与集合X的交集，$S(F, \alpha^0) \equiv \{x : F(x) \leqslant \alpha^0\}$

集中，而且有一个子序列$\{x^{k_i*}\}$收敛于极限$\hat{x}\in X$。设$\hat{x}$不是问题的最优点，则

$$F(\hat{x})>F(x^*)$$

序列$\{F(x^{k_i*})+t(\rho_{k_i})q(x^{k_i*})-F(x^*)\}$就不收敛于零，产生矛盾。所以

$$\lim_{k\to\infty}\{P(x^{k*},\ \rho_k)\}=F(x^*)$$

也就是必须$F(\hat{x})=F(x^*)$，即$\hat{x}$就是问题的最优解。证毕。

在附有等式约束的情况下，内点法是不能直接采用的，这时要改用惩罚函数，即

$$P(x,\ \rho_k,\ \eta_k)=F(x)+t(\rho_k)q(x)+r(\eta_k)S(x)$$

按上述条件，一般也能证明出这种序列无约束极小化问题解的极限就是带约束极小化问题的解。

第五章　无约束极小化的解析法

§1　一元函数的极小化

从以上几章所讨论的最优化问题，我们知道，通过SUMT方法或者拉格朗日乘子方法，以及其它的解法，都是有约束的极小化问题，转化为求解无约束的极小化问题。

一般求解无约束极小化问题

$$极小化F(x),\quad x\in D\subset R^n \tag{5-1}$$

可以包括解析法和直接法两种。解析法：主要是通过求导数或与导数有关的分析得到极小点 x^*；直接法：是不用导数而直接从式（5-1）的函数形式，或者甚至是从离散的实验数据进行比较逐渐逼近地求得极小点 x^*的近似数值。

如果一个实变量 x 对应给出的实函数是：

$$y=F(x),\quad x\in D\subset R^1 \tag{5-2}$$

我们现在先来探讨这种一元函数的极小化问题，即

$$极小化\quad F(x),\quad x\in D\subset R^1 \tag{5-3}$$

关于这种映射

$$F:D\subset R^1\to R^1 \tag{5-4}$$

的最优化问题，就是要在 R^1空间中 D域内求出一点 x^*，使 $F(x)$ 达极小值，也就是说，求极小点 $x^*\in D$ 使

$$F(x^*)\leqslant F(x),\quad \forall x\in D\subset R^1 \tag{5-5}$$

严格地说，能够使式（5-5）成立的 x^*称为$F(x)$ 在D上的一个整体极小点；或者如果有：

$$F(x^*)<F(x),\quad \forall x\in D\subset R^1 \tag{5-6}$$

则称 x^*是$F(x)$ 在D上的一个整体严格极小点。但是，更多的

情况是我们只能在D中找到一个x^*的邻域，也就是在一维空间中只能定出x^*的附近的开区间（a，b），或更一般的统用符号$N(x^*)$一个邻域使

$$F(x^*)\leqslant F(x),\quad \forall x\in N(x^*)\in R^1 \tag{5-7}$$

或

$$F(x^*)<F(x),\quad \forall x\in N(x^*)\in R^1 \tag{5-8}$$

则称x^*是$F(x)$在D上的局部极小点或局部严格极小点。

设一元函数$F(x)$在D中有二阶连续导数，将$F(x)$在D的内点$\hat{x}$展成泰勒展开，即

$$F(x)=F(\hat{x})+F'(\hat{x})(x-\hat{x})+\frac{1}{2}F''(\hat{x})(x-\hat{x})^2+o(\Delta x^2) \tag{5-9}$$

其中：$\Delta x=x-\hat{x}$，式（5-9）总能在$\hat{x}$的一个邻域$N(\hat{x})$成立。

如果$\hat{x}=x^*$是$F(x)$的一个极小点，则从式（5-9）展成一次项的

$$F(x)=F(x^*)+F'(x^*)(x-x^*)+o(\Delta x^2)$$

中，可知x^*是$F(x)$极小点的必要条件是：

$$F'(x^*)=0 \tag{5-10}$$

若写成：

$$F'(x)=f(x) \tag{5-11}$$

则当$F(x)$的极小点x^*确实存在时，我们就可以一般的从

$$f(x)=0 \tag{5-12}$$

这个方程求根而得。另外，在必要条件（5-10）成立以后，再由（5-9）式得：

$$F(x)=F(x^*)+\frac{1}{2}F''(x^*)(x-x^*)^2+o(\Delta x^2)$$

$$\Delta x=x-x^*$$

由此可见：

$$F'(x^*)=0,\quad F''(x^*)>0 \tag{5-13}$$

是x^*为$F(x)$局部严格极小点（或称局部强极小点）

$$F(x)>F(x^*),\quad \forall x\in N(x^*)\subset D\subset R^1 \qquad (5\text{-}14)$$

的充分而必要的条件。

当$F(x)$是一个线性函数

$$F(x)=ax+b,\quad a\neq 0$$

时，由于$F'(x)=a\neq 0$，所以没有极小值的问题。当然，也没有什么极大值的问题。要一个线性函数发生最优化的要求，必然是带有一些约束条件的。

当$F(x)$是一个二次函数

$$F(x)=ax^2+bx+c,\quad a\neq 0$$

时，从

$$F'(x)=2ax+b=0$$

解出逗留点（可能是极点）

$$\hat{x}=-\frac{b}{2a}$$

如果

$$F''(x)=2a>0$$

也就是在$a\neq 0$，$a>0$时，$x=-\dfrac{b}{2a}$就是二次函数一个局部强极小点，而且也是一个整体强极小点。如果所给二次函数对所有x有定义，则将是：

$$F(x)>F(x^*),\quad \forall x\in R^1$$

一般来说，关于非线性函数$F(x)$，若已知$F''(x)>0$，又能从$F'(x)=0$的$f(x)=0$解出x^*，那么x^*就是$F(x)$的一个局部强极小点，甚至可能是$F(x)$的一个整体强极小点。$F(x^*)$就是所要求的一个最优值。

一般的求解非线性方程（5-12）是困难的，我们就将用线性函数$l(x)$逐次的在极小点x^*的附近的近似点x^k代替$f(x)$，然后用$l(x)=0$求解代替$f(x)=0$求解，做进一步接近x^*，可能有比较x^k较好的x^{k+1}，然后，再在x^{k+1}点用另一个线性函数$l(x)$代替$f(x)$，再用$l(x)=0$代替$f(x)=0$求解，

以得到再进一步的改进点 x^{k+2}，如此下去，直到充分接近 x^* 的某一个 x^n 为止。

首先，我们从 $y=f(x)$ 的图象来看，使 $f(x)=0$ 成立的点就是 $y=f(x)$ 曲线和横坐标轴 x 轴的交点，见图5-1。设我们已经从 x^0 初始点开始逐步接近了 x^*，在第 k 步以后，得到 x^k 点，我们可以从 $(x^k, f(x^k))$ 点做曲线 $y=f(x)$ 的切线，交于 x 轴给出 x^{k+1} 点。这个意思用解析式来说，就是采用切线近似函数的一种线性函数：

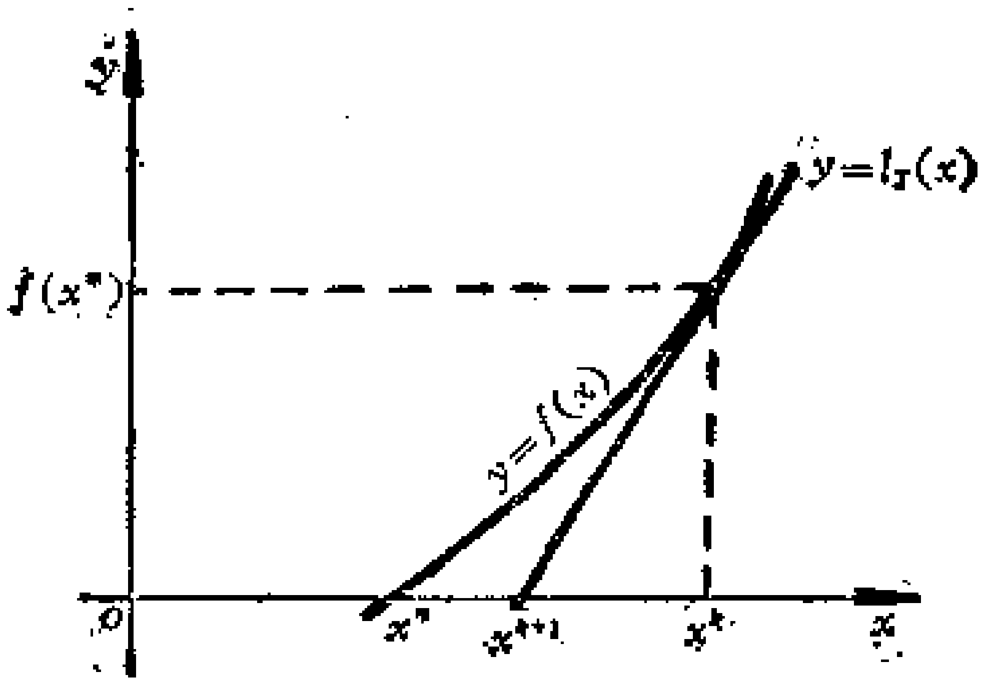


图5-1 切线线性函数的逼近

$$l_T(x)=\alpha x+\beta \tag{5-15}$$

再用 x^k 来代替 $f(x)$，要求在 x^k 点函数值和导数值相等，即

$$l_T(x^k)=f(x^k),\quad l_T'(x^k)=f'(x^k) \tag{5-16}$$

从而，由

$$\alpha x^k+\beta=f(x^k),\quad \alpha=f'(x^k)$$

就得到：

$$\begin{aligned}l_T(x)&=f'(x^k)x+f(x^k)-f'(x^k)x^k\\&=f(x^k)+f'(x^k)(x-x^k)\end{aligned}$$

现在，$f'(x^k)\neq 0$（否则 $x^k=x^*$），所以由 $l_T(x)=0$ 解得：

$$x^{k+1}=x^k-\frac{f(x^k)}{f'(x^k)} \tag{5-17}$$

$$k=0,1,2,\cdots$$

这里的 $k=0,1,2,\cdots$，就说明我们从一开始就采用了这种切线近似的线性函数所解得的迭代关系。式（5-17）就是著名的牛顿-雷夫生-康托罗维奇（Newton-Raphson-Канторович）迭代公式，简称为牛顿迭代公式。

如果按原来的目标函数 $F(x)$ 写出牛顿迭代公式就是：

$$x^{k+1}=x^k-[F''(x^k)]^{-1}F'(x^k) \tag{5-18}$$

$$k=0,1,2,\cdots$$

牛顿迭代方法式（5-17）或式（5-18）是一般迭代方法

$$x^{k+1}=\varphi(x^k) \tag{5-19}$$

的一种。

一般的情况，我们可以把式（5-12）的非线性方程

$$f(x)=0 \tag{5-12}$$

写成：

$$x=\varphi(x)$$

按式（5-19）进行迭代，这种迭代能收敛于（5-12）方程根的充分条件是：$\varphi(x)$ 有一阶连续导数，且

$$|\varphi'(x)|\leqslant\alpha<1,\quad \forall x\in|x-x^*|<\delta \tag{5-20}$$

其中：x^*是式（5-12）的零点，也就是 $F(x)$ 的极小点。

因为，从 $|x-x^*|<\delta$ 的 $F(x)$ 有定义的区间 (a,b) 内任取二点 x'，x''，就有：

$$\varphi(x')-\varphi(x'')=\int_{x''}^{x'}\varphi'(x)dx$$

所以有：

$$|\varphi(x')-\varphi(x'')|\leqslant\left|\int_{x''}^{x'}|\varphi'(x)|dx\right|$$

$$\leqslant\{\max|\varphi'(x)|\}|x'-x''|$$

由式（5-20）就可以知道，$\varphi(x)$ 是满足李普希兹（Lipschitz）条件，即

$$|\varphi(x')-\varphi(x'')|\leqslant\alpha|x'-x''|,\quad 0<\alpha<1 \tag{5-21}$$

这样，我们就有了相继的比较关系式：

$$|x^2-x^1|=|\varphi(x^1)-\varphi(x^0)|\leqslant\alpha|x^1-x^0|$$

$$|x^3-x^2|=|\varphi(x^2)-\varphi(x^1)|\leqslant\alpha|x^2-x^1|\leqslant\alpha^2|x^1-x^0|$$

从归纳法，可得到：

$$|x^{k+1}-x^k|\leqslant\alpha^k|x^1-x^0|$$

因此，级数

$$x^0+(x^1-x^0)+(x^2-x^1)+\cdots+(x^{k+1}-x^k)+\cdots$$

各项的绝对值都小于级数

$$|x^0|+|x^1-x^0|+\alpha|x^2-x^1|+\cdots+\alpha^k|x^{k+1}-x^k|+\cdots$$

的对应项。后者在 $0<\alpha<1$ 时是一个收敛级数，因此，前一级数的首 $k+2$ 项之和收敛，也就是：

$$\lim_{n\to\infty}x^n=x^*$$

同时，从 $\varphi(x)$ 的连续性知道

$$\lim_{k\to\infty}x^{k+1}=\lim_{k\to\infty}\varphi(x^k)=\varphi(\lim_{k\to\infty}x^k)=\varphi(x^*)$$

也就是有"不动点"的 x^* 关系，即

$$x^*=\varphi(x^*)$$

所以，x^* 是通过 $x=\varphi(x)$ 的迭代过程 $x^{k+1}=\varphi(x^k)$ 求出的 $f(x)=0$ 的根，也就是 $F(x)$ 的极小点。

这样，求出的根也是唯一的，因为如果有 x^* 和 $\bar{x}^*$ 使：

$$x^*=\varphi(x^*)\qquad\qquad\bar{x}^*=\varphi(\bar{x}^*)$$

则

$$0<|x^*-\bar{x}^*|=|\varphi(x^*)-\varphi(\bar{x}^*)|\leqslant\alpha|x^*-\bar{x}^*|$$

$$0<\alpha<1$$

这是不可能的，只有

$$x^*=\bar{x}^*$$

才可能。

在牛顿迭代法（5-17）中有：

$$\varphi(x)=x-\frac{f(x)}{f'(x)}$$

所以，收敛于不动点(也就是极小点)x^*的充分条件是：

$$|\varphi'(x)|=\left|\frac{f(x)f''(x)}{[f'(x)]^2}\right|\leqslant\alpha<1 \qquad (5\text{-}22)$$

并因

$$\varphi'(x)=\frac{f(x)f''(x)}{[f'(x)]^2}$$

$$\varphi''(x)=\frac{[f'(x)f''(x)+f(x)f'''(x)]f'(x)-2f(x)[f''(x)]^2}{[f'(x)]^3}$$

所以有：

$$\varphi'(x^*)=0,\quad \varphi''(x^*)=\frac{f''(x^*)}{f'(x^*)}$$

分析牛顿迭代的第 k 步，将由展开给出估计，即

$$\varphi(x^k)=\varphi(x^*)+\frac{1}{2}\,\frac{f''(x^*)}{f'(x^*)}(x^k-x^*)^2+o(\Delta x^3)$$

其中：$\Delta x=x^k-x^*$。所以能找到 c，使

$$|x^{k+1}-x^*|=|\varphi(x^k)-\varphi(x^*)|\leqslant c\,|x^k-x^*|^2 \qquad (5\text{-}23)$$

这就表明，使用牛顿迭代算法，一旦误差很小，x^k十分接近x^*以后，平方阶就将快速收敛，这是牛顿迭代法的最大优点。

虽然牛顿迭代法在收敛的吸引范围内收敛很快，这是它的优点，但是，使用牛顿迭代方法，要进一步求出$f(x)$的导数，这就在实际步骤中增加了很多工作，这是它的缺点。为了避免对$f(x)$再次求导，一种办法是用差商代替$f(x)$的导数，或者说是用$y=f(x)$图示曲线上相邻二点间的割线代替一点的切线，从而问题就是把在曲线一点和该点切线斜率用曲线上二点代替，也就是用函数$f(x)$在二点上的值代替函数在一点上的值和在该点的函数导数值，见图 5-2。

从算法上来说，现在改用线性函数

$$l(x)=\alpha_1 x+\beta_1 \qquad (5\text{-}24)$$

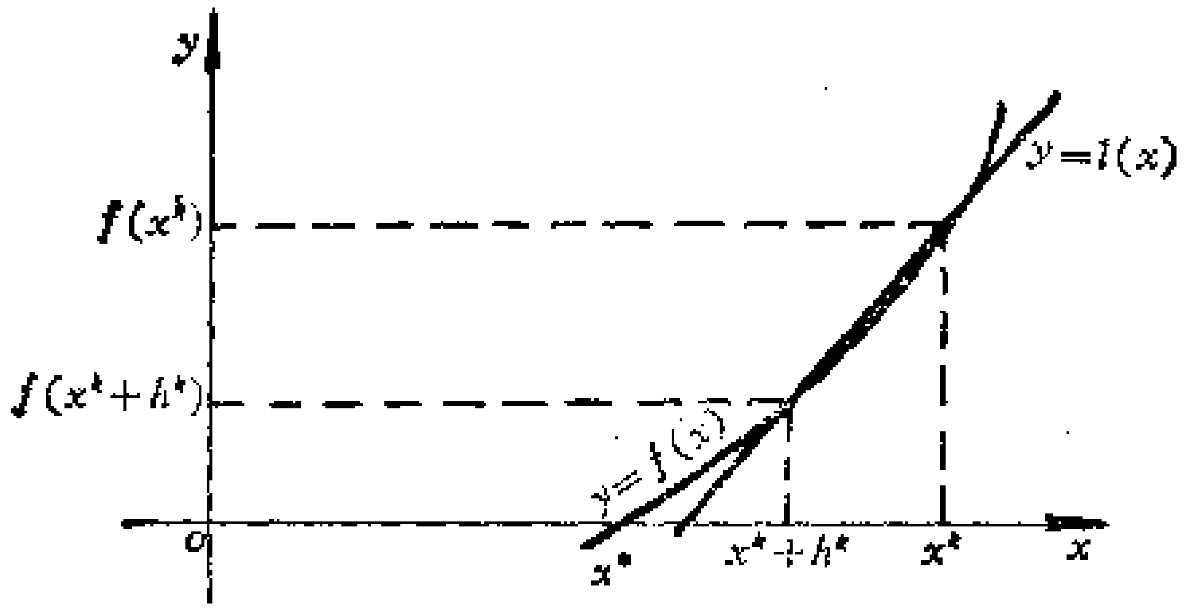


图5-2 用割线代替切线的图示

代替式（5-15），用来表达在 x^k 的近似函数，最后逼近 $f(x)$。这时取 x^k 附近一点 x^k+h^k 使

$$l(x^k)=f(x^k),\quad l(x^k+h^k)=f(x^k+h^k)$$

则式（5-24）的 α_1，β_1 可以从

$$\alpha_1 x^k+\beta_1=f(x^k),\quad \alpha_1(x^k+h^k)+\beta_1=f(x^k+h^k)$$

解出为：

$$\alpha_1=\frac{f(x^k+h^k)-f(x^k)}{h^k},\quad \beta_1=f(x^k)-\alpha_1 x^k$$

所以

$$l(x)=f(x^k)+\frac{f(x^k+h^k)-f(x^k)}{h^k}(x-x^k)$$

然后，用 $l(x)=0$ 代替在 x^k 附近解 $f(x)=0$，得到的进一步解值为：

$$x^{k+1}=x^k-\left[\frac{f(x^k+h^k)-f(x^k)}{h^k}\right]^{-1}f(x^k) \tag{5-25}$$

这就是将牛顿迭代公式离散化改型以后的割线迭代公式。它在 h^k 相当小的情况下，几乎可以代替牛顿迭代公式，而且如果把式（5-25）写成迭代关系：

$$x^{k+1}=x^k-J(f,\ h^k)^{-1}f(x^k) \tag{5-26}$$

$$k=0,1,2,\cdots$$

则式（5-26）具有相容逼近性质，即

$$\lim_{h^k\to 0}J(f,\ h^k)=f'(x^k)$$

我们的目的是按照这样的思路导出多元函数非线性方程的牛顿迭代公式和它的离散改型。这在实用上是极为重要的。

§2 多元函数的极小化

把上面关于一元函数求极小化的问题，完全平行的来处理多元函数的极小化，我们就可以比较容易地找到多元函数非线性方程的各种迭代方法，并且还可以进一步探讨它们之间的关系。

如果把一个 n 元实变量写成列向量：

$$\mathbf{x}=(x_1, \ x_2, \ \cdots, \ x_n)^T$$

它给出的实函数是：

$$y=F(\mathbf{x}), \quad \mathbf{x}\in D\subset R^n, \quad F(\mathbf{x})\in R^1 \tag{5-27}$$

或者用映射写成：

$$F: \ D\subset R^n\to R^1 \tag{5-28}$$

所谓 n 元函数 $F(\mathbf{x})$ 的极小化，就是要在 D 中找出一点 $\mathbf{x}^*\in D\subset R^n$ 使在 $\mathbf{x}^*$ 的一个邻域 $N(\mathbf{x}^*)$ 有局部极小性质，即

$$F(\mathbf{x}^*)\leqslant F(\mathbf{x}), \quad \forall \mathbf{x}\in N(\mathbf{x}^*)\subset D\subset R^n$$

或有局部严格（强）极小性质，即

$$F(\mathbf{x}^*)<F(\mathbf{x}), \quad \forall \mathbf{x}\in N(\mathbf{x}^*)\subset D\subset R^n$$

甚至有一个整体极小性质，即

$$F(\mathbf{x}^*)\leqslant F(\mathbf{x}), \quad \forall \mathbf{x}\in D\subset R^n$$

或一个整体严格（强）极小性质，即

$$F(\mathbf{x}^*)<F(\mathbf{x}), \quad \forall \mathbf{x}\in D\subset R^n$$

若 $F(\mathbf{x})$ 在 $\mathbf{x}\in D$ 内有一阶连续偏导数时，可就 D 中的一个内点

$$\hat{\mathbf{x}}=(\hat{x}_1, \ \hat{x}_2, \ \cdots, \ \hat{x}_n)^T$$

来将 $F(\mathbf{x})$ 展成泰勒展式为：

$$F(\hat{\mathbf{x}}+\varepsilon\mathbf{y})=F(\hat{\mathbf{x}})+\varepsilon\mathbf{y}^T\hat{\mathbf{g}}+o(\varepsilon^2) \tag{5-29}$$

$$\hat{\mathbf{x}}+\varepsilon\mathbf{y}\in D$$

所以，$\hat{\mathbf{x}}=\mathbf{x}^*$ 是 $F(\mathbf{x})$ 极小点的必要条件是对充分小的 ε 实

数，或者说有 x^* 的邻域 $N(x^*)$ 使：

$$F(x^*)\leqslant F(x^*+\varepsilon y),\quad x^*+\varepsilon y\in N(x^*)$$

所以，必须有：

$$y^T g^* = 0 \tag{5-30}$$

也就是 x^* 为 $F(x)$ 极小点的必要条件是 $F(x)$ 在 x^* 点的梯度向量为：

$$g^*=\nabla_x F(x^*)=\left(\frac{\partial F(x^*)}{\partial x_1},\ \frac{\partial F(x^*)}{\partial x_2},\cdots,\frac{\partial F(x^*)}{\partial x_n}\right)^T=0$$

对于多元一次函数

$$F(x)=b^T x+c=b_1x_1+b_2x_2+\cdots+b_nx_n+c,\ x\in R^n$$

$$b=(b_1,\ b_2,\ \cdots,b_n)^T\neq 0$$

的梯度向量

$$\nabla_x F=(b_1,\ b_2,\ \cdots,\ b_n)^T=b\neq 0$$

无所谓极小的问题。如果线性函数再加以一些线性约束条件才构成线性规划问题，求得落在约束边界超平面上的极小点。

对于二次函数

$$F(x)=\frac{1}{2}x^T Gx+b^T x+c$$

$$=\frac{1}{2}\sum_{i,j=1}^{n} g_{ij}x_ix_j+\sum_{i=1}^{n} b_ix_i+c \tag{5-31}$$

$$x\in R^n$$

其中：b，$x\in R^n$，$G\in L(R^n)$ 是一个对称矩阵。

我们已知 $b^T x+c$ 的梯度是 b，而 $\frac{1}{2}x^T Gx$ 这个二次型的梯度向度正是 Gx，因为将二次型对任何变量求导时，有：

$$\left(\frac{1}{2}x^T Gx\right)'=\frac{1}{2}\sum_{i,j=1}^{n} g_{ij}x_i'x_j+\frac{1}{2}\sum_{i,j=1}^{n} g_{ij}x_ix_j'$$

$$=\sum_{i,j=1}^{n} g_{ij}x_i'x_j=\sum_{i,j=1}^{n} g_{ij}x_ix_j'$$

$$=(x')^T Gx=x^T G(x')$$

特别是对 x 的第 i 个分量求导，就有：

$$\left(\frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x}\right)'_{x_i}=(0,0,\cdots,0,1,0,\cdots,0)\mathbf{G}\mathbf{x}=(\mathbf{G}\mathbf{x})_i$$

其中：第一个行向量只有第 i 个元为 1，其余都是零。而 $(\mathbf{G}\mathbf{x})_i$ 表示向量 $\mathbf{G}\mathbf{x}$ 的第 i 个元。所以有：

$$\nabla_x\left(\frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x}\right)=((\mathbf{G}\mathbf{x})_1,\ (\mathbf{G}\mathbf{x})_2,\ \cdots,\ (\mathbf{G}\mathbf{x})_n)^T=\mathbf{G}\mathbf{x}$$

也就是得到一般二次函数（5-31）的梯度函数是：

$$\nabla_x F=\mathbf{G}\mathbf{x}+\mathbf{b} \tag{5-32}$$

所以，对于二次函数（5-31），只要对称的方阵 G 是非奇的，就可以解得二次函数唯一的逗留点，即

$$\hat{\mathbf{x}}=-\mathbf{G}^{-1}\mathbf{b} \tag{5-33}$$

现在，就这个逗留点将二次函数写成泰勒展式，就是：

$$F(\hat{\mathbf{x}}+\varepsilon\mathbf{y})=F(\hat{\mathbf{x}})+\varepsilon\mathbf{y}^T\hat{\mathbf{g}}+\frac{1}{2}\varepsilon^2\mathbf{y}^T\mathbf{G}\mathbf{y}=F(\hat{\mathbf{x}})+\frac{1}{2}\varepsilon^2\mathbf{y}^T\mathbf{G}\mathbf{y}$$

可见，只要 G 是一个正定矩阵，$\hat{\mathbf{x}}$ 就是所求的一个整体强极小点 $\mathbf{x}^*$。

例 1. 设给出二次函数

$$F(\mathbf{x})=4x_1^2+x_2^2-40x_1-12x_2+136$$

则配平方得：

$$F(\mathbf{x})=4(x_1-5)^2+(x_2-6)^2$$

由此可知，有一个极小点 $\mathbf{x}^*=(5,6)^T$，并且 $F(\mathbf{x}^*)=0$。也可以用二次型的标准形式把二次函数写成：

$$F(\mathbf{x})=\frac{1}{2}(x_1,\ x_2)\begin{pmatrix}8 & 0\\ 0 & 2\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}+\begin{pmatrix}-40\\ -12\end{pmatrix}^T\begin{pmatrix}x_1\\ x_2\end{pmatrix}+136$$

这样就得到：

$$\mathbf{G}=\begin{pmatrix}8 & 0\\ 0 & 2\end{pmatrix},\ \mathbf{b}=\begin{pmatrix}-40\\ -12\end{pmatrix},\quad c=136$$

并且 G 是一个正定矩阵，于是从梯度函数等于零的方程：

$$\mathbf{G}\mathbf{x}+\mathbf{b}=0$$

即

$$\begin{pmatrix}8 & 0\\0 & 3\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}+\begin{pmatrix}-40\\-12\end{pmatrix}=0$$

解出：

$$\begin{pmatrix}x_1\\x_2\end{pmatrix}=-\begin{pmatrix}8 & 0\\0 & 2\end{pmatrix}^{-1}\begin{pmatrix}-40\\-12\end{pmatrix}=\frac{1}{16}\begin{pmatrix}2 & 0\\0 & 8\end{pmatrix}\begin{pmatrix}40\\12\end{pmatrix}$$

$$=\frac{1}{16}\begin{pmatrix}8 & 0\\9 & 6\end{pmatrix}=\begin{pmatrix}5\\6\end{pmatrix}$$

关于一般非线性函数的映射（见式（5-28）），即

$$F: D\subset R^n\to R^1$$

如果$F(\mathbf{x})$在D内有连续的二阶偏导数，$\hat{\mathbf{x}}\in D$是D的一个内点，就可以有$\hat{\mathbf{x}}$的一个邻域$N(\hat{\mathbf{x}})$将$F(\mathbf{x})$展成：

$$F(\hat{\mathbf{x}}+\varepsilon\mathbf{y})=F(\hat{\mathbf{x}})+\varepsilon\mathbf{y}^T\hat{\mathbf{g}}+\frac{1}{2}\varepsilon^2\mathbf{y}^T\hat{\mathbf{G}}\mathbf{y}+o(\varepsilon^3) \qquad (5\text{-}34)$$

$$\hat{\mathbf{x}}+\varepsilon\mathbf{y}\in N(\hat{\mathbf{x}})$$

其中：$\hat{\mathbf{g}}$是$F(\mathbf{x})$在$\hat{\mathbf{x}}$的梯度函数，写成：

$$\hat{\mathbf{g}}=\nabla_x F(\hat{\mathbf{x}})=f(\hat{\mathbf{x}}) \qquad (5\text{-}35)$$

$\hat{\mathbf{G}}$是$F(\mathbf{x})$在$\hat{\mathbf{x}}$的赫森（Hessian）矩阵：

$$\hat{\mathbf{G}}=\mathbf{G}(\hat{\mathbf{x}})=H_F(\hat{\mathbf{x}})=\begin{pmatrix}\dfrac{\partial^2 F}{\partial x_1^2} & \dfrac{\partial^2 F}{\partial x_1\partial x_2} & \cdots & \dfrac{\partial^2 F}{\partial x_1\partial x_n}\\ \dfrac{\partial^2 F}{\partial x_2\partial x_1} & \dfrac{\partial^2 F}{\partial x_2^2} & \cdots & \dfrac{\partial^2 F}{\partial x_2\partial x_n}\\ \cdots & \cdots & \cdots & \cdots\\ \dfrac{\partial^2 F}{\partial x_n\partial x_1} & \dfrac{\partial^2 F}{\partial x_n\partial x_2} & \cdots & \dfrac{\partial^2 F}{\partial x_n^2}\end{pmatrix}_{\mathbf{x}=\hat{\mathbf{x}}} \qquad (5\text{-}36)$$

若在$\hat{\mathbf{x}}=\mathbf{x}^*$时，则

$$\mathbf{g}^*=\nabla_x F(\mathbf{x}^*)=f(\mathbf{x}^*)=0$$

并且

$$\mathbf{G}(\mathbf{x}^*)=H_F(\mathbf{x}^*)$$

是一个正定矩阵，则 $\mathbf{x}^*$就是$F(\mathbf{x})$的一个局部强极小点。

因此，一般在已知$F(\mathbf{x})$有一极小点存在时，我们就要求解多元非线性方程

$$f(\mathbf{x})=0 \tag{5-37}$$

的根。求解的办法和一元函数的情况一样，设从某一接近极小点 $\mathbf{x}^*$的初始点 $\mathbf{x}^0$开始，在第 k 步，若已经得到 $\mathbf{x}^k$ 是一个比较接近 $\mathbf{x}^*$的点，我们可以在 $\mathbf{x}^k$ 用一个多元线性函数来近似 $f(\mathbf{x})$，设它是：

$$L_T(\mathbf{x})=\mathbf{A}_1\mathbf{x}+\mathbf{b}^{(1)} \tag{5-38}$$

其中：$\mathbf{b}^{(1)}\in R^n$ 是一个 n 向量，$\mathbf{A}_1\in L(R^n)$ 是一个 n 阶方阵，因此式（5-38）共有 n^2+n 个待定参数。

现在，完全类似于一元函数方程导出牛顿迭代公式那样，我们可以采用下式：

$$f(\mathbf{x}^k)=\left(\frac{\partial F(\mathbf{x}^k)}{\partial x_1},\ \frac{\partial F(\mathbf{x}^k)}{\partial x_2},\ \cdots,\ \frac{\partial F(\mathbf{x}^k)}{\partial x_n}\right)^T \tag{5-39}$$

$$f'(\mathbf{x}^k)=\begin{pmatrix} \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_1^2} & \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_1\partial x_2} & \cdots & \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_1\partial x_n} \\ \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_2\partial x_1} & \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_2^2} & \cdots & \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_2\partial x_n} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_n\partial x_1} & \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_n\partial x_2} & \cdots & \dfrac{\partial^2 F(\mathbf{x}^k)}{\partial x_n^2} \end{pmatrix} \tag{5-40}$$

这样，用已知的 n^2+n 个值来确定 $\mathbf{A}_1$和 $\mathbf{b}^{(1)}$。但还应注意，式（5-40）给出的 $f(\mathbf{x})$ 的伽杜导数是我们在第一章中已经介绍过了的，也就是这里的

$$f'(\mathbf{x})=(\nabla_x F)'=H_F(\mathbf{x}) \tag{5-41}$$

是梯度向量函数 $\nabla_x F$ 的伽杜导数也就是给出$F(\mathbf{x})$的赫森函数矩阵。而赫森函数矩阵就是 $f(\mathbf{x})$ 的伽杜导数。

我们令

$$L_T(\mathbf{x}^k)=f(\mathbf{x}^k),\quad L_T'(\mathbf{x}^k)=f'(\mathbf{x}^k) \tag{5-42}$$

而因为 $L_T(\mathbf{x})$ 是一个 n 列向量函数，它的各个分量是：

$$[L_T(\mathbf{x})]_i=\sum_{j=1}^{n}a_{ij}x_j+\mathbf{b}^{(i)}$$

所以

$$L_T'(\mathbf{x})=\begin{pmatrix}\frac{\partial[L_T(\mathbf{x})]_1}{\partial x_1} & \frac{\partial[L_T(\mathbf{x})]_1}{\partial x_2} & \cdots & \frac{\partial[L_T(\mathbf{x})]_1}{\partial x_n}\\ \frac{\partial[L_T(\mathbf{x})]_2}{\partial x_1} & \frac{\partial[L_T(\mathbf{x})]_2}{\partial x_2} & \cdots & \frac{\partial[L_T(\mathbf{x})]_2}{\partial x_n}\\ \cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots\\ \frac{\partial[L_T(\mathbf{x})]_n}{\partial x_1} & \frac{\partial[L_T(\mathbf{x})]_n}{\partial x_2} & \cdots & \frac{\partial[L_T(\mathbf{x})]_n}{\partial x_n}\end{pmatrix}$$

$$=\begin{pmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ a_{n1} & a_{n2} & \cdots & a_{nn}\end{pmatrix}=\mathbf{A}$$

因此有：

$$\mathbf{A}_1=H_F(\mathbf{x}^k),\quad \mathbf{A}_1\mathbf{x}^k+\mathbf{b}^{(1)}=f(\mathbf{x}^k)$$

从而得到：

$$L_T(\mathbf{x})=f(\mathbf{x}^k)+H_F(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k)$$

可见，如果在 $\mathbf{x}^*$的附近能保持 $H_F(\mathbf{x})$ 正定时，我们就可以用

$$L_T(\mathbf{x})=0$$

近似代替 $f(\mathbf{x})=0$，解出：

$$\mathbf{x}^{k+1}=\mathbf{x}^k-[H_F(\mathbf{x}^k)]^{-1}f(\mathbf{x}^k) \tag{5-43}$$

所以，只要保持 $H_F(\mathbf{x})$ 正定，我们就有了多元非线性方程(5-37) 的牛顿迭代公式，即

$$\mathbf{x}^{k+1}=\mathbf{x}^k-[f'(\mathbf{x}^k)]^{-1}f(\mathbf{x}^k) \tag{5-44}$$

$$k=0,1,2,\cdots$$

或者

$$x^{k+1}=x^k-[H_F(x^k)]^{-1}\nabla_x F(x^k) \qquad (5\text{-}45)$$

$$k=0,1,2,\cdots$$

或者，有时把F的赫森矩阵看成是f的雅可比矩阵，就可以写出牛顿迭代公式是：

$$x^{k+1}=x^k-[J(f,\ x^k)]^{-1}f(x^k) \qquad (5\text{-}46)$$

$$k=0,1,2,\cdots$$

其中

$$J(f,\ x^k)=\begin{pmatrix} \frac{\partial f_1(x^k)}{\partial x_1} & \frac{\partial f_1(x^k)}{\partial x_2} & \cdots & \frac{\partial f_1(x^k)}{\partial x_n} \\ \frac{\partial f_2(x^k)}{\partial x_1} & \frac{\partial f_2(x^k)}{\partial x_2} & \cdots & \frac{\partial f_2(x^k)}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n(x^k)}{\partial x_1} & \frac{\partial f_n(x^k)}{\partial x_2} & \cdots & \frac{\partial f_n(x^k)}{\partial x_n} \end{pmatrix}$$

$f_1(x)$，$f_2(x)$，…，$f_n(x)$ 是$f(x)$ 的n个分量函数。

例 2　仍以例 1 的二次函数$F(x)=4x_1^2+x_2^2-40x_1-12x_2+136$为例，这时它的梯度函数是：

$$\nabla_x F(x)=f(x)=\begin{pmatrix} 8x_1-40 \\ 2x_2-12 \end{pmatrix}$$

赫森矩阵函数是：

$$H_F(x)=f'(x)=\begin{pmatrix} \frac{\partial(8x_1-40)}{\partial x_1} & \frac{\partial(8x_1-40)}{\partial x_2} \\ \frac{\partial(2x_2-12)}{\partial x_1} & \frac{\partial(2x_2-12)}{\partial x_2} \end{pmatrix}=\begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix}$$

这就是例 1 中的 G，它恒是一个常值矩阵。本例的等值曲线如图5-3 所示，若现在取初值点是 $x^0=(8,9)^T$，则用牛顿迭代公式就可以一次的求得极小值是：

$$\mathbf{x}^1=\mathbf{x}^0-\begin{pmatrix}8 & 0\\ 0 & 2\end{pmatrix}^{-1}\begin{pmatrix}8x_1-40\\ 2x_2-12\end{pmatrix}_{(8,9)}=\mathbf{x}^0-\begin{pmatrix}8 & 0\\ 0 & 2\end{pmatrix}^{-1}\begin{pmatrix}24\\ 6\end{pmatrix}$$

$$=\begin{pmatrix}8\\ 9\end{pmatrix}-\begin{pmatrix}\frac{1}{8} & 0\\ 0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}24\\ 6\end{pmatrix}=\begin{pmatrix}8\\ 9\end{pmatrix}-\begin{pmatrix}3\\ 3\end{pmatrix}=\begin{pmatrix}5\\ 6\end{pmatrix}$$

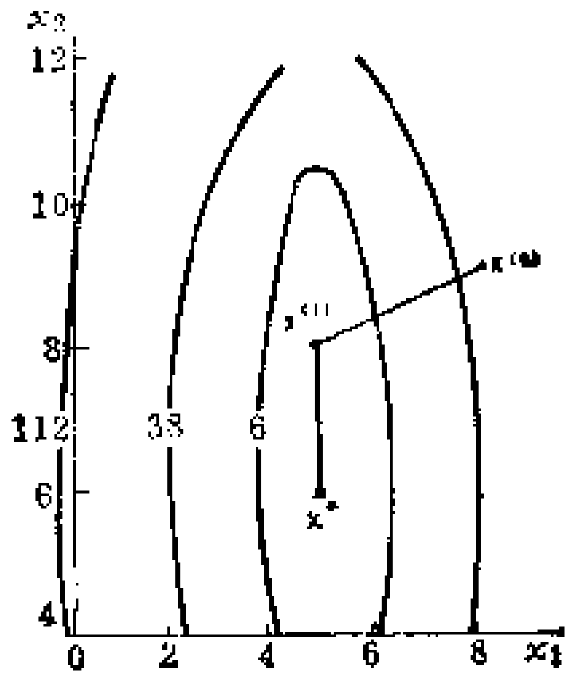


图5-3 二次函数的等值曲线

由此可见，对于二次函数，无论初值点取为何值，总是用牛顿迭代一次得到极小点的。这是因为二次函数$F(\mathbf{x})$的梯度函数$f(\mathbf{x})$本身就是线性函数，而且

$$f(\mathbf{x})=\mathbf{G}\mathbf{x}+\mathbf{b}=\mathbf{G}(\mathbf{x}^k+\mathbf{x}-\mathbf{x}^k)+\mathbf{b}=\mathbf{G}\mathbf{x}^k+\mathbf{b}+\mathbf{G}(\mathbf{x}-\mathbf{x}^k)$$
$$=f(\mathbf{x}^k)+\mathbf{G}(\mathbf{x}-\mathbf{x}^k)=L_T(\mathbf{x})$$

所以$L_T(\mathbf{x})=0$和$f(\mathbf{x})=0$同样的步骤可得解。

§3 牛顿迭代法的收敛性

我们在探讨一元函数方程（5-12）的迭代求解问题时，把

$$f(x)=0$$

写成

$$x=\varphi(x)$$

使用迭代

$$x^{k+1}=\varphi(x^k)\qquad k=0,1,2,\cdots$$

的过程，并且通过式（5-20）

$$|\varphi'(x)|\leqslant\alpha<1,\quad\forall x\in|x-x^*|<\delta$$

的条件给出 $\varphi(x)$ 的李普希兹条件性质（式（5-21））为：

$$|\varphi(x')-\varphi(x'')|\leqslant\alpha|x'-x''|\qquad 0<\alpha<1$$

这样就决定了牛顿迭代的收敛性。

现在，将多元函数方程（5-37）转换为：

$$\mathbf{x}=\Phi(\mathbf{x})\tag{5-47}$$

针对这个映射 $\Phi(\mathbf{x})$，若能找到 $\mathbf{x}^*$ 使式（5-47）成立，即

$$\mathbf{x}^*=\Phi(\mathbf{x}^*)\tag{5-48}$$

这个不动点 $\mathbf{x}^*$ 就是我们要求的式（5-37）的根，也就是 $F(\mathbf{x})$ 的极小点。

可以证明，只要

$$\mathbf{x}=\Phi(\mathbf{x}),\quad \mathbf{x}\in D\subset R^n,\quad \Phi(\mathbf{x})\in D\subset R^n$$

而 D 是一个 n 维的完备向量子空间，而且具有压缩映射性质：

对 D 中任何二元 $\mathbf{x}'$，$\mathbf{x}''$ 就有某给定范数使式

$$\|\Phi(\mathbf{x}')-\Phi(\mathbf{x}'')\|\leqslant\alpha\|\mathbf{x}'-\mathbf{x}''\|\tag{5-49}$$

$$0<\alpha<1$$

成立，则必在 D 中有唯一的定点 $\mathbf{x}^*$ 使式（5-48）成立。

我们是从构造上来证明这个压缩映射原理的。

如果 $\mathbf{x}^0$ 是 D 中一元，且

$$\mathbf{x}^1=\Phi(\mathbf{x}^0)$$
$$\mathbf{x}^2=\Phi(\mathbf{x}^1)$$
$$\cdots\quad\cdots\quad\cdots$$
$$\mathbf{x}^k=\Phi(\mathbf{x}^{k-1})$$

则从 Φ 的压缩性质，有：

$$\|\mathbf{x}^n-\mathbf{x}^m\|=\|\Phi(\mathbf{x}^{n-1})-\Phi(\mathbf{x}^{m-1})\|\leqslant\alpha\|\mathbf{x}^{n-1}-\mathbf{x}^{m-1}\|$$

若设 $m>n$，就有：

$$\|\mathbf{x}^n-\mathbf{x}^m\|\leqslant\alpha^n\|\mathbf{x}^0-\mathbf{x}^{m-n}\|$$

但是

$$\|x^0-x^{m-n}\|=\|x^0-x^1+x^1-x^2+x^2-\cdots+x^{m-n+1}-x^{m-n}\|$$
$$\leqslant\|x^0-x^1\|+\|x^1-x^2\|+\cdots+\|x^{m-n+1}-x^{m-n}\|$$
$$\leqslant\|x^0-x^1\|(1+\alpha+\alpha^2+\cdots+\alpha^{m-n+1})$$
$$\leqslant\|x^0-x^1\|\cdot\frac{1}{1-\alpha}$$

所以有：

$$\|x^n-x^m\|\leqslant\frac{\alpha^n}{1-\alpha}\|x^0-x^1\|$$

可见$\{x^n\}$是一个柯西基本序列，由于D是完备的，所以总有 x^* 使下式成立：

$$\lim_{n\to\infty}\|x^n-x^*\|=0$$

另一方面，从压缩映射性质式（5-49）可知，Φ是在D上连续的，所以

$$\Phi(x^*)=\lim_{n\to\infty}\Phi(x^n)=\lim_{n\to\infty}x^{n+1}=x^*$$

并且 x^* 这个吸引点还是唯一的，因为如果另有一个$\bar{x}^*$使

$$\bar{x}^*=\Phi(\bar{x}^*)$$

则从

$$\|\bar{x}^*-x^*\|=\|\Phi(\bar{x}^*)-\Phi(x^*)\|\leqslant\alpha\|\bar{x}^*-x^*\|$$

可证，只有$\bar{x}^*=x^*$才有可能。压缩映射原理证毕。

从这个构造上证明压缩映射原理的过程来看，说明一旦我们把问题经过三转化步序，即

$$\min F(x)\Longrightarrow f(x)=0\Longrightarrow x=\Phi(x) \qquad (5\text{-}50)$$

只要$\Phi(x)$具有压缩映射性质，就可以有下列收敛迭代的过程：

$$x^{k+1}=\Phi(x^k) \qquad k=0,\ 1,\ 2,\ \cdots \qquad (5\text{-}51)$$

而所求问题的极小点 x^* 就是迭代（5-51）的收敛吸引点。

1966 年奥斯托斯基（Ostrowski）给出关于吸引点的一个重要定理，指出：

设 $\Phi: D\subset R^n\to R^n$ 有一个定点 $x^*\in \mathrm{INT}(D)$ 且在 x^* 伽杜可微，若 $\Phi'(x^*)$ 的谱半径合于 $\rho(\Phi'(x^*))=\sigma<1$，则 x^* 是迭代（5-51）的一个吸引点。

这里 $\mathrm{INT}(D)$ 表示 D 的内点集合，谱半径 $\rho(\Phi'(x^*))$ 表示 $\Phi'(x^*)$ 的特征值绝对值的最大值。

特别是由牛顿型的映射

$$\Phi: D\subset R^n\to R^n$$

$$\Phi(x)=x-A(x)^{-1}f(x),\quad \forall\ x\in D \tag{5-52}$$

还可以证明：

$$\Phi'(x^*)=I-A(x^*)^{-1}f'(x^*) \tag{5-53}$$

所以，就一般牛顿型迭代：

$$x^{k+1}=x^k-A(x^k)^{-1}f(x^k) \tag{5-54}$$

$$k=0,\ 1,\ 2,\ \cdots$$

奥氏定理的条件就是：

$$\rho(I-A(x^*)^{-1}f'(x^*))=\sigma<1 \tag{5-55}$$

就特殊的牛顿迭代而言，此时：

$$\Phi(x)=x-H_F(x)^{-1}f(x)$$

条件是：

$$\rho(I-H_F(x^*)^{-1}f'(x^*))=\sigma<1$$

而 $f'(x^*)=H_F(x^*)$。所以在 $H_F(x)$ 正定时，恒有 x^* 的吸引域使牛顿迭代收敛于定点 x^*。

例 3　最优化问题中一个典型例子是极小化罗森布罗克（Rosenbrock）函数　$F(x)=100(x_2-x_1^2)^2+(1-x_1)^2$。

它的极小点显然是 $x^*=(1,\ 1)^T$，并且 $F(1,\ 1)=0$。但是它的等值线很奇特，见图 5-4，初始点选得较远时，收敛的步序很慢。

例如：取　$x^0=(-0.5,\ 0.5)^T$ 时，则：

$$F(x^0)=8.5$$

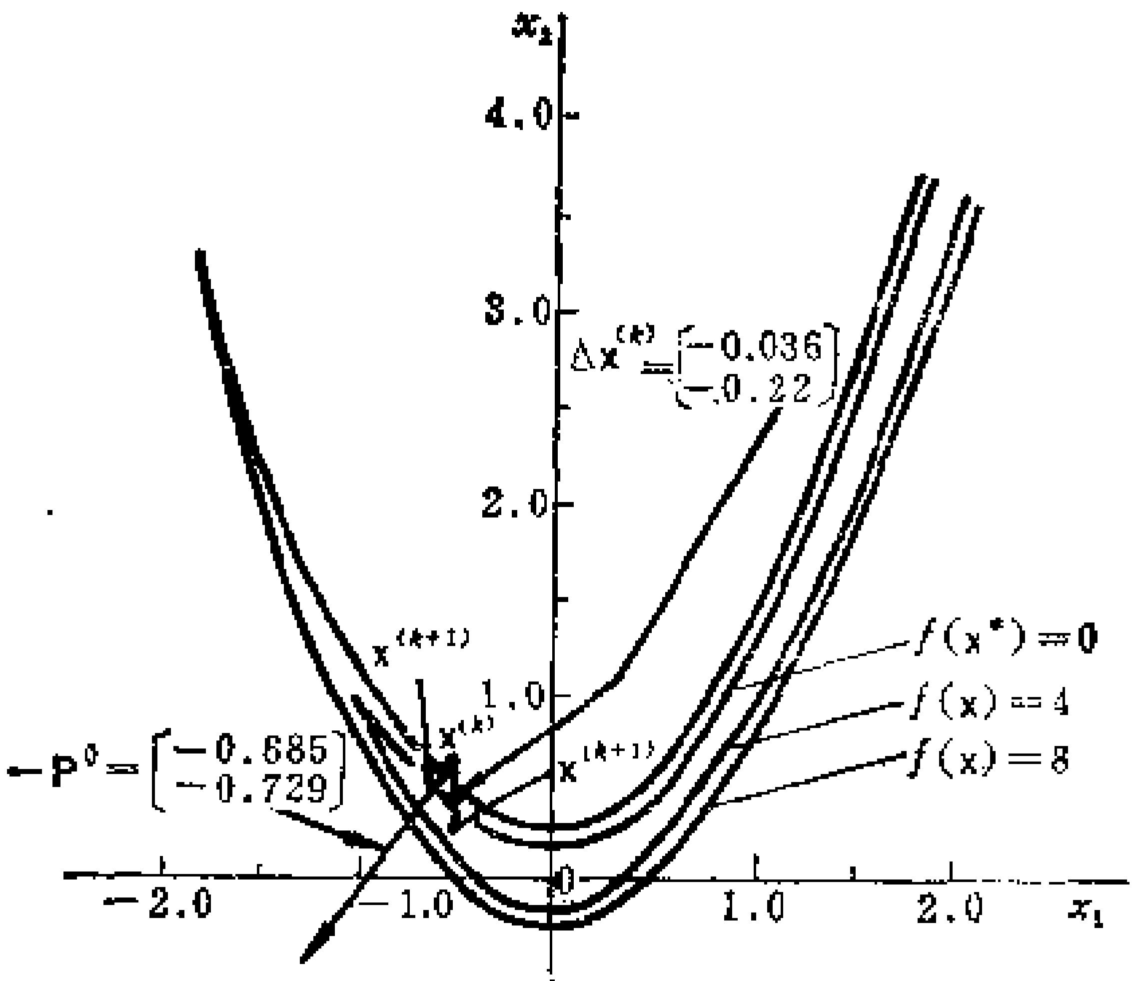


图5-4　罗森布罗克函数的等值曲线

$$f(\mathbf{x}^0)=\begin{pmatrix}-200(x_2-x_1^2)(2x_1)-2(1-x_1)\\200(x_2-x_1^2)\end{pmatrix}_{\mathbf{x}^0}=\begin{pmatrix}47\\50\end{pmatrix}$$

$$H_F(\mathbf{x}^0)=\begin{pmatrix}-400x_2+1200x_1^2+2 & -400x_1\\-400x_1 & 200\end{pmatrix}_{\mathbf{x}^0}=\begin{pmatrix}102 & 200\\200 & 200\end{pmatrix}$$

$$H_F(\mathbf{x}^0)^{-1}=\frac{1}{\begin{vmatrix}102 & 200\\200 & 200\end{vmatrix}}\begin{pmatrix}200 & -200\\-200 & 102\end{pmatrix}=\frac{1}{98}\begin{pmatrix}-1 & 1\\1 & -0.51\end{pmatrix}$$

所以，经过一次牛顿迭代，得到：

$$\mathbf{x}^1=\begin{pmatrix}-0.5\\0.5\end{pmatrix}-\frac{1}{98}\begin{pmatrix}-1 & 1\\1 & -0.51\end{pmatrix}\begin{pmatrix}47\\50\end{pmatrix}=\begin{pmatrix}-0.53\\0.28\end{pmatrix}$$

并且

$$F(\mathbf{x}^1)=F(-0.53,\ 0.28)=2.33$$

它表示，一方面迭代所得的点比原来的 $\mathbf{x}^0$ 点较近一些 $\mathbf{x}^*$，同时函数值也大大地减小了。

但是，使用牛顿迭代法主要是点的逐渐接近于 $\mathbf{x}^*$，而不必一

定每一步的函数值保证减小。例如：取$x^0=(0, 0)^T$时，

$$F(x^0)=1$$

$$f(x^0)=\begin{pmatrix}-2\\0\end{pmatrix}$$

$$H_F(x^0)=\begin{pmatrix}2 & 0\\0 & 200\end{pmatrix}$$

所以就有：

$$x^1=\begin{pmatrix}0\\0\end{pmatrix}-\begin{pmatrix}\frac{1}{2} & 0\\0 & \frac{1}{200}\end{pmatrix}\begin{pmatrix}-2\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$$

$$F(x^1)=100$$

这时 x^1 点更接近于 x^*，但 $F(x^1)$ 却远远大于 $F(x^0)$。

§4 牛顿迭代法的离散化改型

多元函数非线性方程的牛顿迭代公式，可以完全类似于一元函数非线性方程牛顿迭代公式用割切迭代公式离散化改型那样，通过 x^k 点附近再选出 n 个点构成 $n+1$ 个点：

$$x^k,\quad x^{k,1},\quad x^{k,2},\quad \cdots,\quad x^{k,n} \tag{5-56}$$

对应的定出 $f(x)$ 的 $n+1$ 个 n 向量值：

$$f(x^k),\quad f(x^{k,1}),\quad f(x^{k,2}),\quad \cdots,\quad f(x^{k,n})$$

简写成：

$$y^k,\quad y^{k,1},\quad y^{k,2},\quad \cdots,\quad y^{k,n} \tag{5-57}$$

它就提供了 n^2+n 个已给值。因此，我们可以在 x^k 用另一个线性向量函数：

$$L(x)=Ax+b \tag{5-58}$$

以改变式（5-38）的线性向量函数。这时，令

$$
\begin{aligned}
&L(\mathbf{x}^k)=\mathbf{A}\mathbf{x}^k+\mathbf{b}=f(\mathbf{x}^k)=\mathbf{y}^k\\
&L(\mathbf{x}^{k,1})=\mathbf{A}\mathbf{x}^{k,1}+\mathbf{b}=f(\mathbf{x}^{k,1})=\mathbf{y}^{k,1}\\
&L(\mathbf{x}^{k,2})=\mathbf{A}\mathbf{x}^{k,2}+\mathbf{b}=f(\mathbf{x}^{k,2})=\mathbf{y}^{k,2}\\
&\cdots\cdots\cdots\cdots\cdots\cdots\\
&L(\mathbf{x}^{k,n})=\mathbf{A}\mathbf{x}^{k,n}+\mathbf{b}=f(\mathbf{x}^{k,n})=\mathbf{y}^{k,n}
\end{aligned}
\tag{5-59}
$$

如果所取的式（5-56）$n+1$ 个点和对应的函数值（5-57）是一般位置，也就是说，

$$\mathbf{x}^{k,1}-\mathbf{x}^k,\quad \mathbf{x}^{k,2}-\mathbf{x}^k,\quad \cdots,\quad \mathbf{x}^{k,n}-\mathbf{x}^k \tag{5-60}$$

或

$$\mathbf{x}^k-\mathbf{x}^{k,1},\quad \mathbf{x}^{k,1}-\mathbf{x}^{k,2},\quad \cdots,\quad \mathbf{x}^{k,n}-\mathbf{x}^{k,n-1} \tag{5-61}$$

是线性独立的，同时

$$\mathbf{y}^{k,1}-\mathbf{y}^k,\quad \mathbf{y}^{k,2}-\mathbf{y}^k,\quad \cdots,\quad \mathbf{y}^{k,n}-\mathbf{y}^k \tag{5-62}$$

或

$$\mathbf{y}^k-\mathbf{y}^{k,1},\quad \mathbf{y}^{k,1}-\mathbf{y}^{k,2},\quad \cdots,\quad \mathbf{y}^{k,n}-\mathbf{y}^{k,n-1} \tag{5-63}$$

是线性独立的。

设从式（5-60）、（5-62）给出矩阵为：

$$
\begin{aligned}
&\mathbf{H}_k=(\mathbf{x}^{k,1}-\mathbf{x}^k,\ \mathbf{x}^{k,2}-\mathbf{x}^k,\ \cdots,\ \mathbf{x}^{k,n}-\mathbf{x}^k)\\
&\mathbf{\Gamma}_k=(\mathbf{y}^{k,1}-\mathbf{y}^k,\ \mathbf{y}^{k,2}-\mathbf{y}^k,\ \cdots,\ \mathbf{y}^{k,n}-\mathbf{y}^k)
\end{aligned}
\tag{5-64}
$$

由式（5-59）则得到：

$$\mathbf{A}(\mathbf{x}^{k,i}-\mathbf{x}^k)=\mathbf{y}^{k,i}-\mathbf{y}^k \qquad i=1,2,\cdots,n \tag{5-65}$$

合写成矩阵形式是：

$$\mathbf{A}\mathbf{H}_k=\mathbf{\Gamma}_k \tag{5-66}$$

现在，$\mathbf{H}_k$，$\mathbf{\Gamma}_k$ 都有逆阵，所以能解出 $\mathbf{A}$ 是：

$$\mathbf{A}_k=\mathbf{\Gamma}_k\mathbf{H}_k^{-1}$$

并且从式（5-59）第一式又可以计出b，所以由式（5-58）得到：

$$L(\mathbf{x})=f(\mathbf{x}^k)+\mathbf{A}_k(\mathbf{x}-\mathbf{x}^k)$$

用 $L(\mathbf{x})=0$ 代替 $f(\mathbf{x})=0$ 就给出牛顿迭代离散改型的迭代公式：

$$\mathbf{x}^{k+1}=\mathbf{x}^k-\mathbf{A}_k^{-1}f(\mathbf{x}^k) \tag{5-67}$$

$$k=0,\ 1,\ 2,\ \cdots$$

其中

$$A_k^{-1}=H_k\Gamma_k^{-1}$$

1967年歌梯加（Ortega）引进相容逼近[1]，研究了离散改型迭代公式（5-67）和牛顿迭代之间的逼近关系。以后，我们实际上应用改型迭代（5-67）时，还要采用下降法配合进行，关于下降法因离散改型产生的误差问题，1976年布格斯[2]等人有进一步的分析。

有两种满足局部收敛性的二点序列离散改型方法是取：

$$x^{k,i}=x^k+(x_i^{k-1}-x_i^k)e^i \tag{5-68}$$

$$i=1,\ 2,\ \cdots,\ n$$

和

$$x^{k,i}=x^k+\sum_{j=1}^{i}(x_j^{k-1}-x_j^k)e^j \tag{5-69}$$

$$i=1,\ 2,\ \cdots,\ n$$

就前一种来说，若设

$$h_i^k=x_i^{k-1}-x_i^k \qquad i=1,\ 2,\ \cdots,\ n$$

则

$$H_k=(x^{k,1}-x^k,\ x^{k,2}-x^k,\ \cdots,\ x^{k,n}-x^k)$$

$$=\begin{pmatrix} h_1^k & 0 & \cdots & 0 \\ 0 & h_2^k & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & h_n^k \end{pmatrix}$$

特别简单的是取 $h_1^k=h_2^k=\cdots=h_n^k=h^k$ 的情况。

一般的 $n+1$ 点序列情况是取：

$$H_k=(x^k-x^{k-1},\ x^{k-1}-x^{k-2},\ \cdots,\ x^{k-n+1}-x^{k-n})$$

---

[1] 见：J. M. Ortega and W. C. Rheinboldt《Iterative Solution of Nonlinear Equations in Several Variables》，1970.

[2] 见：P. T. Boggs and J. E. Dennis, Jr,《Math. of Comp.》, p. 199~215, 1976.

设令

$$p^i = x^{i+1} - x^i, \quad q^i = f(x^{i+1}) - f(x^i)$$

则有：

$$H_k = (p^{k-1}, \ p^{k-2}, \ \cdots, \ p^{k-n})$$
$$\Gamma_k = (q^{k-1}, \ q^{k-2}, \ \cdots, \ q^{k-n}) \qquad (5\text{-}70)$$

按此，恒有关系式：

$$A_{k+i} H_{k+i} = \Gamma_{k+i} \qquad i = 0, \ 1, \ 2, \ \cdots\cdots \qquad (5\text{-}71)$$

现在，我们来分析一下这样取序列点的性质。

首先，在式（5-71）中对 $i=0$，$i=1$ 的情况，则有：

$$A_k p^j = q^j \qquad j = k-1, k-2, \cdots, k-n+1, \ k-n$$
$$A_{k+1} p^j = q^j \qquad j = k, k-1, k-2, \cdots, k-n+1 \qquad (5\text{-}72)$$

所以合并以后，就有：

$$(A_{k+1} - A_k) p^j = 0 \qquad j = k-1, k-2, \cdots, k-n+1$$
$$A_{k+1} p^k = q^k \qquad (5\text{-}73)$$

而且因为：

$$A_k p^k = -f(x^k) \ \text{即} \ x^{k+1} - x^k = -A_k^{-1} f(x^k)$$

所以

$$(A_{k+1} - A_k) p^k = q^k + f(x^k) = f(x^{k+1})$$

从而式（5-73）的 n 个联立向量方程是：

$$(A_{k+1} - A_k) p^j = 0 \qquad j = k-1, k-2, \cdots, k-n+1$$
$$(A_{k+1} - A_k) p^k = f(x^{k+1}) \qquad (5\text{-}74)$$

但是，p^k，p^{k-1}，p^{k-2}，$\cdots$，p^{k-n+1} 是 n 个线性独立向量，所以可解为：

$$A_{k+1} - A_k = (f(x^{k+1}), 0, 0, \cdots, 0)(p^k, p^{k-1}, \cdots, p^{k-n+1})^{-1}$$

从而 $A_{k+1} - A_k$ 是一个秩 1 的 n 阶方阵。在第一章，我们已经知道，一个秩 1 方阵的充分必要的条件是有 u^k，$v^k \in R^n$ 使下式成立：

$$A_{k+1} - A_k = u^k (v^k)^T \qquad (5\text{-}75)$$

$$k=0,1,\cdots\cdots$$

还有，我们知道著名的SMWH公式（Sherman-Morrison-Woodbury-Householder）为：

$$(A+uv^T)^{-1}=A^{-1}-A^{-1}u(I+v^TA^{-1}u)^{-1}v^TA^{-1} \tag{5-76}$$

所以，从式（5-75），我们就可以导出关于A_k逆A_k^{-1}的递推关系。

因此，在各相应矩阵有逆阵存在的情况下，式（5-75）用式(5-76)改写，则有：

$$A_{k+1}^{-1}=[A_k+u^k(v^k)^T]^{-1}=A_k^{-1}-\frac{A_k^{-1}u^k(v^k)^TA_k^{-1}}{1+(v^k)^TA_k^{-1}u^k} \tag{5-77}$$

这样，我们就得到牛顿迭代离散改型计算公式的递推计算，也就是在确定A_0之后，从式（5-77）我们就可以直接运算式(5-67)，即

$$x^{k+1}=x^k-A_k^{-1}f(x^k)$$

$$k=0,\ 1,\ 2,\cdots$$

如果我们用

$$B_k=A_k^{-1}$$

根据式（5-67）和式（5-77）就有：

$$x^{k+1}=x^k-B_kf(x^k)$$

$$B_{k+1}=B_k-\frac{B_ku^k(v^k)^TB_k}{1+(v^k)^TB_ku^k} \tag{5-78}$$

$$k=0,\ 1,\ 2,\ \cdots$$

一般的，我们可以按

$$A_{k+1}-A_k=U^k(V^k)^T$$

$$U^k,\ V^k\in L(R^m,\ R^n)\qquad m\leqslant n$$

给出，则从SMWH公式就有：

$$\begin{aligned}A_{k+1}^{-1}&=A_k^{-1}-A_k^{-1}U^k[I_m+(V^k)^TA_k^{-1}U^k]^{-1}(V^k)^TA_k^{-1}\\&=A_k^{-1}+Y^k(W^k)^T\end{aligned} \tag{5-79}$$

其中

$$Y^k = -A_k^{-1}U^k[I_m + (V^k)^T A_k^{-1} U^k]^{-1}$$
$$W^k = (A_k^{-1})^T V^k$$

都是 $n \times m$ 矩阵。另外，我们选 Y^k，W^k 构成的 $A_k^{-1} = B_k$ 应该能满足

$$x^{k+1} - x^k = -B_k f(x^k) \Longrightarrow A_k p^k = -f(x^k)$$

也就是应该满足式（5-74）的第二个关系，即

$$(A_{k+1} - A_k)p^k = f(x^{k+1}) \tag{5-80}$$

否则在迭代过程之中就要发生矛盾。

最常用的一种按式（5-79）、（5-80）构造的变尺度迭代公式，采用秩 2 的 Y^k，W^k，也就是说，Y^k，W^k 是各由二个 n 向量所组成：

$$Y^k = (Y^{1,k},\ Y^{2,k}),\quad W^k = (W^{1,k},\ W^{2,k})$$

这时，式（5-80）性质也可以从

$$A_{k+1}p^k = q^k \Longrightarrow B_{k+1}q^k = p^k$$
$$p^k = -B_k f(x^k) = -B_k[f(x^{k+1}) - q^k]$$

给出是：

$$(B_{k+1} - B_k)q^k = -B_k f(x^{k+1}) \tag{5-81}$$

现在式（5-79）是：

$$B_{k+1} - B_k = (Y^{1,k},\ Y^{2,k})\begin{pmatrix}(W^{1,k})^T \\ (W^{2,k})^T\end{pmatrix} = Y^{1,k}(W^{1,k})^T + Y^{2,k}(W^{2,k})^T$$

所以，选 Y^k，W^k 应使

$$Y^{1,k}(W^{1,k})^T q^k + Y^{2,k}(W^{2,k})^T q^k = -B_k f(x^{k+1}) \tag{5-82}$$

能够使式（5-80）成立的一种选法是：

$$Y^{1,k} = \frac{p^k}{(W^{1,k})^T q^k},\quad Y^{2,k} = -\frac{B_k q^k}{(W^{2,k})^T q^k} \tag{5-83}$$

$$(W^{1,k})^T q^k \neq 0,\quad (W^{2,k})^T q^k \neq 0$$

因为将式（5-83）代入式（5-82）以后就是：

$$p^k - B_k q^k = -B_k f(x^{k+1})$$

此式显然是成立的。

这样，我们就得到了一类秩 2 的变尺度迭代公式是：

$$x^{k+1}=x^k-B_k f(x^k) \qquad k=0,\ 1,\ 2,\ \cdots$$

$$B_{k+1}=B_k+\frac{p^k(W^{1,k})^T}{(W^{1,k})^T q^k}-\frac{B_k q^k(W^{2,k})^T}{(W^{2,k})^T q^k} \qquad (5\text{-}84)$$

在这一类秩 2 变尺度的类牛顿迭代公式中，特别是取

$$W^{1,k}=p^k,\quad W^{2,k}=B_k^T q^k$$

就是 1960 年前后导出的著名的 DFP（Davidon-Fletcher-Powell）公式：

$$B_{k+1}=B_k+\frac{p^k(p^k)^T}{(p^k)^T q^k}-\frac{B_k q^k(q^k)^T B_k}{(q^k)^T B_k q^k} \qquad (5\text{-}85)$$

$$k=0,\ 1,\ 2,\ \cdots$$

使用这个公式时，我们可以取 $B_0=I$，所需的点也只是二点序列，它在联合下降法使用时，效果很好。

如果再进一步分析式（5-84）的一般秩 2 变尺度迭代公式中的变尺度阵递推式，可以写成：

$$\begin{aligned}B_{k+1}&=B_k+p^k\frac{(W^{1,k})^T}{(W^{1,k})+q^k}-B_k q^k\frac{(W^{2,k})^T}{(W^{2,k})^T q^k}\\&=B_k+p^k(s^k)^T-B_k q^k(r^k)^T\end{aligned}$$

注意到，这里的 s^k 和 r^k 故有性质，即

$$(s^k)^T q^k=1,\quad (r^k)^T q^k=1$$

所以就可以将 DFP 公式加以单参数修改为：

$$(s^k)^T=\frac{[1+\beta(q^k)^T B_k q^k](p^k)^T}{(p^k)^T q^k}-\beta(q^k)^T B_k$$

$$(r^k)^T=\frac{[1+\beta(p^k)^T q^k](q^k)^T B_k}{(q^k)^T B_k q^k}-\beta(p^k)^T$$

其中：β是待定参数。从而给出加修正项的 DFP 公式改型形式为：

$$B_{k+1}=B_k+\frac{p^k(p^k)^T}{(p^k)^Tq^k}-\frac{B_kq^k(q^k)^TB_k}{(q^k)^TB_kq^k}$$

$$+\beta\left[\frac{B_kq^k(q^k)^TB_k}{(q^k)^TB_kq^k}(p^k)^Tq^k-B_kq^k(p^k)^T-p^k(q^k)^TB_k\right.$$

$$\left.+\frac{p^k(p^k)^T}{(p^k)^Tq^k}(q^k)^TB_kq^k\right] \qquad (5\text{-}86)$$

这里，β参数项的作用使计算上更趋稳定。

在单参数秩 2 的变尺度递推式（5-86）中，在$\beta=0$时就是DFP 公式；在 $\beta=-\dfrac{1}{(p^k)^Tq^k}$时，就是 1970 年前后出现的BFS（Broyden-Fletcher-Shanno）公式；而在$\beta=-1/(q^k)^TB_kq^k$时，就是 1970 年格林斯塔（Greenstadt）提出的公式。

从上面分析看起来，变尺度迭代公式起源于牛顿迭代，所以常称为拟牛顿迭代或类牛顿迭代。牛顿迭代的一个重要特点就是关于二次函数是一步就可以求得极小点的。离散化了的改型迭代所派生出的一系列变尺度迭代显然不可能仍具有一步求得极小点的性质。但是，它们又具备了另一类十分重要的性质，就是针对下降法的方向而言，它们可以形成一系列的关于二次函数的共轭方向。

最一般的能够形成共轭方向的变尺度迭代是 1971 年阿达奇(Adachi)● 给出的修正类。但是比较简单一些的，则是包括相当广泛的黄氏提出的多参数类●，其算式为：

$$B_{k+1}=B_k+\rho\frac{p^k(c_1p^k+c_2B_k^Tq^k)^T}{(c_1p^k+c_2B_k^Tq^k)^Tq^k}-\frac{B_kq^k(k_1p^k+k_2B_k^Tq^k)^T}{(k_1p^k+k_2B_k^Tq^k)^Tq_k} \qquad (5\text{-}87)$$

其中：ρ，c_1，c_2，k_1，k_2是给出三自由族

$$(\rho,\ c_1/c_2,\ k_1/k_2)$$

的参数。

● 见：《JOTA》，Vol. 7 p. 391～423.

● 见：《JOTA》，Vol. 5 p. 405～423.

§5 下 降 法

我们在以上各节分析了非线性方程求根的牛顿迭代方法，以及围绕着牛顿迭代公式的离散化的改型所推导出的一系列变尺度的拟牛顿迭代公式。它们的重要性在于关于一般非线性方程的线性近似，所以它一方面关于二次目标函数，牛顿迭代公式能够一次完成，而另一方面也因为这种针对二次函数的改型，离散化之后的一系列迭代公式，不仅对非线性方程求根提出方便的方法，而且还特别具有一些二次函数适合的特性，例如DFP公式以及其它更多参数的递推变尺度公式都反映了至多 n 步就能得到二次函数极小点的性质。

但是，关于一般非二次函数的目标函数，使用牛顿迭代或一般的变尺度拟牛顿迭代公式，由于初始点选得不好，会收敛得慢，甚至不能收敛。为此，我们在一般迭代公式（5-67），即

$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\mathbf{A}_k^{-1}f(\mathbf{x}^{k}) \qquad k=0,\ 1,\ 2,\ \cdots$$

或式（5-68），即

$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\mathbf{B}_k f(\mathbf{x}^{k}) \qquad k=0,\ 1,\ 2,\ \cdots$$

中增添上一个加快步伐的参数因子 α_k 构成带阻尼作用的迭代公式：

$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\alpha_k\mathbf{B}_k f(\mathbf{x}^{k}) \qquad k=0,\ 1,\ 2,\ \cdots \quad (5\text{-}88)$$

如选取适当的 α_k 步长保证得到的待最优化目标函数下降，这就是：

$$F(\mathbf{x}^{k+1})=F(\mathbf{x}^{k}-\alpha_k\mathbf{B}_k f(\mathbf{x}^{k}))<F(\mathbf{x}^{k})$$

现暂且不究 $\mathbf{x}^{k+1}$ 是否更接近于极小点 $\mathbf{x}^*$，而设法使每一步都做到待优函数下降，在下降到的点足够近 $\mathbf{x}^*$ 时，再使 $\alpha_k=1$，放开步子使收敛加快。

从一般的迭代性质看起来，这种下降迭代的意义就是把每次的

$$B_k f(x^k) \qquad k=0,\ 1,\ 2,\ \cdots$$

当做下降方向

$$p^k = x^{k+1} - x^k$$

并改写为 $-P^k$ 以修改其大小为：

$$p^k = x^{k+1} - x^k = -\alpha_k P^k$$

也就是在迭代过程中，第 k 步采用 P^k 方向的下降迭代，即

$$x^{k+1} = x^k - \alpha_k P^k \qquad k=0,\ 1,\ 2,\ \cdots \tag{5-89}$$

选 α_k 使下式成立：

$$F(x^{k+1}) = F(x^k - \alpha_k P^k) < F(x^k)$$

我们现在来证明关于下降法能够成立的条件。

设 $F:\ D \subset R^n \to R^1$ 是在 $x \in \mathrm{INT}(D)$ 伽杜可导，且对某方向向量 $P \in R^n$ 有：

$$F'(x)P > 0 \tag{5-90}$$

则有 $\delta > 0$ 使 $\forall \alpha \in (0,\ \delta)$ 能保持目标函数 $F(x)$ 的下降关系，即

$$F(x - \alpha P) < F(x) \tag{5-91}$$

证明：

当 $F(x)$ 在 x 点伽杜可导时，对于使式（5-90）成立的 $P \in R^n$，应有：

$$\lim_{t \to 0} \frac{\| F(x + tP) - F(x) - tF'(x)P \|}{t} = 0$$

令 $t = -\alpha$，$\alpha > 0$ 则有：

$$\lim_{\alpha \to 0} \frac{\| F(x - \alpha P) - F(x) + \alpha F'(x)P \|}{\alpha} = 0$$

也就是

$$\lim_{\alpha \to 0} \frac{1}{\alpha} [F(x - \alpha P) - F(x)] + F'(x)P = 0$$

现在 $F'(x)P > 0$，所以可选 ε 使 $F'(x)P > \varepsilon > 0$，而另一方

面，对于这个给定的 ε，又恒有 δ 使当 $\alpha\in(0,\ \delta)$ 时有：

$$\frac{1}{\alpha}[F(\mathbf{x}+\alpha\mathbf{P})-F(\mathbf{x})]+F'(\mathbf{x})\mathbf{P}<\varepsilon$$

也就是

$$\frac{1}{\alpha}[F(\mathbf{x}-\alpha\mathbf{P})-F(\mathbf{x})]+F'(\mathbf{x})\mathbf{P}<F'(\mathbf{x})\mathbf{P}$$

即

$$\frac{1}{\alpha}[F(\mathbf{x}-\alpha\mathbf{P})-F(\mathbf{x})]<0$$

故有 $\delta>0$ 使 $\forall\alpha\in(0,\ \delta)$ 能有：

$$F(\mathbf{x}-\alpha\mathbf{P})<F(\mathbf{x})$$

证毕。

所以，在下降迭代式（5-89）中，一旦在第 k 步序中，能有 $\mathbf{P}^k$ 方向使

$$F'(\mathbf{x}^k)\mathbf{P}^k>0$$

就恒可选定适当的 $\alpha_k>0$，使迭代

$$\mathbf{x}^{k+1}=\mathbf{x}^k-\alpha_k\mathbf{P}^k$$

能有

$$F(\mathbf{x}^{k+1})<F(\mathbf{x}^k)$$

的结果。

最简单的一种下降法是在

$$F'(\mathbf{x}^k)^T\neq0$$

时，取

$$\mathbf{P}^k=F'(\mathbf{x}^k)^T=\nabla_x F(\mathbf{x}^k)\tag{5-92}$$

这时，因为

$$F'(\mathbf{x}^k)\mathbf{P}^k=F'(\mathbf{x}^k)F'(\mathbf{x}^k)^T=\|F'(\mathbf{x}^k)^T\|^2>0$$

其中：范数就是普通的欧氏范数，它是由梯度向量 $\nabla_x F(\mathbf{x}^k)$ 的各个分量的平方之和的平方根构成。另外，如果 $\nabla_x F(\mathbf{x}^k)=0$，则 $\mathbf{x}^k=\mathbf{x}^*$ 已经无须再进行迭代。所以在取方向是式（5-92）的时候，就得到梯度下降法：

$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\alpha_k F'(\mathbf{x}^{k})^T$$

或
$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\alpha_k \nabla_x F(\mathbf{x}^{k})$$

或
$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\alpha_k f(\mathbf{x}^{k}) \tag{5-93}$$
$$k=0,\ 1,\ 2,\ \cdots$$

每次，我们选 α_k 使下式成立：

$$\min_{\alpha} F(\mathbf{x}^k-\alpha\mathbf{P}^k)=F(\mathbf{x}^k-\alpha_k\mathbf{P}^k)$$

在梯度下降法中，我们也可以每次选单位梯度方向向量

$$\mathbf{P}^k=\frac{\nabla_x F(\mathbf{x}^k)}{\|\nabla_x F(\mathbf{x}^k)\|}$$

形成下降迭代

$$\mathbf{x}^{k+1}=\mathbf{x}^{k}-\alpha_k\frac{\nabla_x F(\mathbf{x}^k)}{\|\nabla_x F(\mathbf{x}^k)\|} \tag{5-94}$$
$$k=0,\ 1,\ 2,\ \cdots$$

例 4 我们再就例 3 的罗森布罗克函数

$$F(\mathbf{x})=100(x_2-x_1^2)^2+(1-x_1)^2$$

用梯度下降迭代式(5-94)，仍按例 3 取初始点 $\mathbf{x}^0=(-0.5,0.5)^T$，$F(\mathbf{x}^0)=8.5$，则从

$$\mathbf{P}^0=\frac{\nabla_x F(\mathbf{x}^0)}{\|\nabla_x F(\mathbf{x}^0)\|}=\frac{1}{\left[\left(\frac{\partial F}{\partial x_1}\right)^2+\left(\frac{\partial F}{\partial x_2}\right)^2\right]^{\frac{1}{2}}_{\mathbf{x}^0}}\begin{pmatrix}\frac{\partial F}{\partial x_1}\\ \frac{\partial F}{\partial x_2}\end{pmatrix}_{\mathbf{x}^0}$$

$$=\frac{1}{68.6}\begin{pmatrix}47\\ 50\end{pmatrix}=\begin{pmatrix}0.685\\ 0.729\end{pmatrix}$$

就可计算出第一次迭代点应是：

$$\mathbf{x}^1=\begin{pmatrix}-0.5\\ 0.5\end{pmatrix}-\alpha_0\begin{pmatrix}0.685\\ 0.729\end{pmatrix}$$

而

$$F(\mathbf{x}^1)=100[0.5-0.729\alpha_0-(0.5+0.685\alpha_0)^2]^2+(1.5+0.685\alpha_0)^2=\hat{F}(\alpha_0)$$

求 $$\min_{\alpha_0} \hat{F}(\alpha_0)$$

得 $$\alpha_0 = 0.164$$

所以，最后得到第一次迭代结果是：

$$x^1 = (-0.612,\ 0.381)^T,\quad F(x^1) = 2.6$$

这个结果并没有我们在例3中用牛顿迭代一次所得的结果好，由此也可以看出牛顿迭代法的优点，这是因为梯度方向只能保证在很短很短的步长之下，它的下降最快，在取步长 α_k 以后，它的总下降量未必都是比较别的迭代强。牛顿迭代法中的赫森矩阵的逆实际上已经比较梯度刻划得更进一步，它把函数 $F(x)$ 在 x^k 点附近的二次导数变化状态都考虑进来了，所以，有时牛顿迭代的效果从下降的意义上来说，也是好的。

但是，牛顿迭代法总的来说，它并不是一种下降法，而是通过目标函数的二阶导数的刻划寻找较快收敛于极小点的一种迭代算法。按理，收敛于极小点的迭代应该经常是也能保证下降的，只是因为牛顿迭代法，它在取定方向方面是好的，即将梯度方向 $f(x^k)$ 改为 $[H_F(x^k)]^{-1} f(x^k)$。但是，所取的步长经常保持 $\alpha = 1$，步长跨大了些，所以，要使牛顿迭代方法也恒能保持下降性，我们就应该在方向之前乘一个阻尼因子，采用下降迭代过程：

$$x^{k+1} = x^k - \alpha_k [H_F(x^k)]^{-1} f(x^k) \tag{5-95}$$

$$k = 0,\ 1,\ 2,\ \cdots$$

这种阻尼牛顿法和梯度下降法（5-93）的内在区别是阻尼牛顿法除了给出对梯度 $f(x^k) = \nabla_x F(x^k)$ 的阻尼因子 α_k 以外，还另外对梯度给出一个预先松弛的因子：

$$\omega_k = [H_F(x^k)]^{-1}$$

这个松弛矩阵使梯度法改型为包括牛顿法在内的，包括变尺度法在内的一般下降迭代：

$$x^{k+1} = x^k - \alpha_k \omega_k f(x^k) \tag{5-96}$$

$$k = 0,\ 1,\ 2,\ \cdots$$

只要这时的 ω_k 是一个正定矩阵，总能选出 α_k 使 $F(x^{k+1})<F(x^k)$，因为设方向

$$P^k=\omega_k f(x^k)=\omega_k F'(x^k)^T$$

则

$$F'(x^k)P^k=F'(x^k)\omega_k F'(x^k)^T>0$$

所以式（5-96）就是一般的下降迭代公式。

特别是阻尼牛顿迭代（5-95），它的赫森矩阵是正定的，其逆也将是正定的，所以式（5-95）就是一种下降迭代。

关于变尺度迭代，我们都可以在变尺度矩阵 B_k 是正定的情况下构成下降法，因为此时有：

$$P^k=F'(x^k)B_k F'(x^k)^T>0$$

特别是在变尺度迭代方法中，最常用的是我们在上面曾介绍的 DFP 公式，即

$$B_{k+1}=B_k+\frac{p^k(p^k)^T}{(p^k)^T q^k}-\frac{B_k q^k(B_k q^k)^T}{(q^k)^T B_k q^k} \tag{5-85}$$

$$B_0=I,\quad p^k=x^{k+1}-x^k,\quad q^k=f(x^{k+1})-f(x^k)$$

$$x^{k+1}=x^k-\alpha_k B_k f(x^k)$$

$$k=0,1,2,\cdots$$

这套公式在 1960 年前后形成之后，经过十年的实践，1970 年前后很多使用者指出在极小化的过程中，用 DFP 下降所需时间最少，并且当使用牛顿迭代法发生赫森矩阵病态时，它也能保持比较好的效果。

事实上，保持式（5-85）是正定的条件是很少的，只要

$$(p^k)^T q^k>0 \tag{5-97}$$

就可以了。

因为式（5-85）中 I 是正定的，设 B 正定，可证：

$$\hat{B}=B+\frac{pp^T}{p^T q}-\frac{(Bq)(Bq)^T}{q^T Bq}$$

也是正定，这里$\hat{B}=B_{k+1}$，$B=B_k$，$p=p^k$，$q=q^k$。而实际上则是：

$$x^T\hat{B}x=x^T\left[B+\frac{pp^T}{p^Tq}-\frac{(Bq)(Bq)^T}{q^TBq}\right]x$$

$$=x^TBx+\frac{(p^Tx)^2}{p^Tq}-\frac{(x^TBq)^2}{q^TBq}$$

$$=\frac{(x^TBx)(q^TBq)-(x^TBq)^2}{q^TBq}+\frac{(p^Tx)^2}{p^Tq}$$

令$y=B^{1/2}x$，$z=B^{1/2}q$，则得：

$$x^T\hat{B}x=\frac{(y^Ty)(z^Tz)-(y^Tz)^2}{z^Tz}+\frac{(p^Tx)^2}{p^Tq}\qquad(5\text{-}98)$$

从舒瓦兹不等式，即

$$(y,z)^2\leqslant(y,y)(z,z)$$

或

$$|(y,z)|\leqslant\|y\|_2\|z\|_2$$

就有：

$$(y^Tz)^2\leqslant(y^Ty)(z^Tz)$$

所以，在$p^Tq>0$时，式（5-98）的右边大于零，也就是说，在B_k正定时，B_{k+1}也必是正定的。

例 5　以例 1 和例 2 的二次函数为例，使用DFP下降法求极小点。

现在待极小的函数是：

$$F(x)=4(x_1-5)^2+(x_2-6)^2$$

它在x_k的梯度是：

$$\nabla_x F(x^k)=\begin{pmatrix}8(x_1-5)\\2(x_2-6)\end{pmatrix}_{x^k}$$

若$x^0=(8,9)^T$是初始迭代点，设$B_0=I$，则DFP的下降迭代，首先要计出α_0使x^1给出的$F(x^1)$下降最大。

现在，从

$$\begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} - \alpha_0 \mathbf{B}_0 \nabla_x F(\mathbf{x}^0) = \begin{pmatrix} 8 \\ 9 \end{pmatrix} - \alpha_0 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 24 \\ 6 \end{pmatrix}$$

也就是用 $x_1^1 = 8 - 24\alpha_0$，$x_2^1 = 9 - 6\alpha_0$ 代入 $F(\mathbf{x})$ 得：

$$F(\mathbf{x}^1) = 4[(8 - 24\alpha_0) - 5]^2 + [(9 - 6\alpha_0) - 6]^2$$

选 α_0 使 $F(\mathbf{x}^1)$ 极小，即

$$\frac{dF(\mathbf{x}^1)}{d\alpha_0} = 0 = 51 - 390\alpha_0$$

得

$$\alpha_0 = 0.1307$$

解出

$$\mathbf{x}^1 = (4.862,\ 8.215)^T$$
$$F(\mathbf{x}^1) = 4.985$$

这里的结果较之 $F(\mathbf{x}^0) = 45$ 是下降了，而且是按 $\mathbf{B}_0 \nabla_x F(\mathbf{x}^0)$ 最大下降的。

再从

$$f(\mathbf{x}^1) = \nabla_x F(\mathbf{x}^1) = (-1.108,\ 4.431)^T$$

进一步求出 $\mathbf{B}_1$。

先计算出 $\Delta f(\mathbf{x}^0) = \mathbf{q}^0$，它是：

$$\mathbf{q}^0 = f(\mathbf{x}^1) - f(\mathbf{x}^0) = \begin{pmatrix} -1.108 \\ 4.431 \end{pmatrix} - \begin{pmatrix} 24 \\ 6 \end{pmatrix} = \begin{pmatrix} -25.108 \\ -1.569 \end{pmatrix}$$

代入 DFP 公式得：

$$\mathbf{B}_1 = \mathbf{B}_0 + \frac{\mathbf{p}^0(\mathbf{p}^0)^T}{(\mathbf{p}^0)^T \mathbf{q}^0} - \frac{\mathbf{B}_0 \mathbf{q}^0 (\mathbf{q}^0)^T \mathbf{B}_0^T}{(\mathbf{q}^0)^T \mathbf{B}_0 \mathbf{q}^0}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{\begin{pmatrix} -3.13 \\ -0.785 \end{pmatrix}(-3.13,\ -0.785)}{(-3.13,\ -0.785)\begin{pmatrix} -25.108 \\ -1.569 \end{pmatrix}}$$

$$-\frac{\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}-25.108\\1.569\end{pmatrix}(-25.108, 1.569)\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}}{(-25.108, -1.569)\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}-25.108\\-1.569\end{pmatrix}}$$

$$=\begin{pmatrix}1.270\times10^{-1} & -3.149\times10^{-2}\\-3.149\times10^{-2} & 1.0038\end{pmatrix}$$

所以，由第一步的迭代出的变尺度给出第二步迭代为：

$$\begin{pmatrix}x_1^2\\x_2^2\end{pmatrix}=\begin{pmatrix}4.862\\8.215\end{pmatrix}-\alpha_1\begin{pmatrix}1.270\times10^{-1} & -3.149\times10^{-2}\\-3.149\times10^{-2} & 1.0038\end{pmatrix}$$

$$\times\begin{pmatrix}-1.108\\4.431\end{pmatrix}$$

因此，可以按上法求出最好的 α_1 是 0.4942，给出二次函数的精确极小点 $\mathbf{x}^2=(5.000,\ 6.000)^T=\mathbf{x}^*$。

若再做一步，仍为 $\mathbf{x}^*$，如下表所示：

第 k 步	x_1^k	x_2^k	$f_1(\mathbf{x}^k)$	$f_2(\mathbf{x}^k)$	$F(\mathbf{x}^k)$	α_k
0	8.000	9.000	24.000	6.000	45.000	0.1307
1	4.862	8.215	−1.108	4.431	4.985	0.4942
2	5.000	6.000	3.81×10^{-7}	2.55×10^{-9}	9.06×10^{-15}	1.000
3	5.000	6.000	0	0	0	—

§6 共轭方向法

从上节例 2 可以看出，关于二次函数，使用牛顿迭代，一次就可以得到极小点，无论所采用的初始点是什么值。

另一方面，从上节例 5 可以看出，关于二次函数，采用 DFP 下降法，只要进行二次迭代就得到极小点。

事实上，我们可以证明只要所用的方向是对于二次函数

$$F(\mathbf{x})=\frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x}+\mathbf{b}^T\mathbf{x}+c,\quad \mathbf{x}\in R^n$$

中 G 是共轭的，即

$$(P^i)^T G P^j = 0 \qquad i \neq j \tag{5-99}$$

时，并按

$$F(x^k - \alpha_k P^k) = \min_{\alpha}\{F(x^k - \alpha P^k) | x^k - \alpha P^k \in R^n\}$$

决定 α_k 时，至多 n 步就可以下降迭代到唯一的极小点 x^*。

因为，此时

$$x^{k+1} = x^k - \alpha_k P^k \qquad k = 0, 1, 2, \cdots, n-1$$

$$F(x^k - \alpha_k P^k) = \min_{\alpha}\{F(x^k - \alpha P^k) | x^k - \alpha P^k \in R^n\}$$

从

$$F'(x^k - \alpha P^k) P^k = 0$$

得知：

$$[G(x^k - \alpha P^k) + b]^T P^k = 0$$

其中：$\nabla_x F = Gx + b$，$F'(x) = (Gx + b)^T$，所以有：

$$\alpha_k = (Gx^k + b)^T P^k / (P^k)^T G P^k \tag{5-100}$$

现在，已有：

$$x^{k+1} = x^k - \frac{(Gx^k + b)^T P^k}{(P^k)^T G P^k} P^k$$

所以

$$(Gx^{k+1} + b)^T P^j = (Gx^k + b)^T P^j - \left[\frac{(Gx^k + b)^T P^k}{(P^k)^T G P^k}\right](P^k)^T G P^j$$

$$0 \leqslant j \leqslant n-1$$

从 P^k 的共轭正交性（5-99），就知道有：

$$(Gx^{k+1} + b)^T P^j = \begin{cases} (Gx^k + b)^T P^j & j \neq k \\ 0 & j = k \end{cases}$$

这样，我们就在第 n 步后得知：

$$(Gx^n + b)^T P^j = (Gx^{n-1} + b)^T P^j = \cdots = (Gx^{j+1} + b)^T P^j = 0$$

$$j = 0, 1, \cdots, n-1$$

由于$P^j(j=0,1,2,\cdots,n-1)$的线性独立性，可知：

$$Gx^n+b=0$$

也就是$x^n=x^*$。

例 6　极小化$F(x)=x_1^2+x_2^2-4$

今取$x^0=(4,4)^T$，并且取方向$P^0=\begin{pmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{pmatrix}$

则

$$x^1=\begin{pmatrix}4\\4\end{pmatrix}-\alpha_0\begin{pmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{pmatrix}$$

而

$$G=\begin{pmatrix}2&0\\0&2\end{pmatrix},\quad \nabla_x F(x^0)=\begin{pmatrix}2&0\\0&2\end{pmatrix}\begin{pmatrix}4\\4\end{pmatrix}=\begin{pmatrix}8\\8\end{pmatrix}$$

所以式（5-100）是：

$$\alpha_0=\frac{(8,8)\begin{pmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{pmatrix}}{\left(\frac{1}{2},\ \frac{\sqrt{3}}{2}\right)\begin{pmatrix}2&0\\0&2\end{pmatrix}\begin{pmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{pmatrix}}=5.46$$

从而得出：$x^1=(1.27,\ -0.74)^T$

其次，再找一个和P^0共轭正交的方向P^1使

$$(P_1^1,\ P_2^1)\begin{pmatrix}2&0\\0&2\end{pmatrix}\begin{pmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{pmatrix}=0$$

并正规化

$$(P_1^1,\ P_2^1)\begin{pmatrix} P_1^1 \\ P_2^1 \end{pmatrix}=1$$

则得：

$$P_1^1=\begin{pmatrix} -\dfrac{\sqrt{3}}{2} \\ \dfrac{1}{2} \end{pmatrix},\ \text{欲使}\alpha_1>0,\ \text{取}\ P_2^1=\begin{pmatrix} \dfrac{\sqrt{3}}{2} \\ -\dfrac{1}{2} \end{pmatrix}$$

又得：

$$\alpha_1=\frac{(2.54,\ -1.48)\begin{pmatrix} \dfrac{\sqrt{3}}{2} \\ -\dfrac{1}{2} \end{pmatrix}}{\left(\dfrac{\sqrt{3}}{2},\ -\dfrac{1}{2}\right)\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} \dfrac{\sqrt{3}}{2} \\ -\dfrac{1}{2} \end{pmatrix}}=1.47$$

所以

$$x^2=\begin{pmatrix} 1.27 \\ -0.74 \end{pmatrix}-1.47\begin{pmatrix} \dfrac{\sqrt{3}}{2} \\ -\dfrac{1}{2} \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

x^2 就是所求 $x^*=(0,0)^T$，见图 5-5。

我们在例 5 中使用 DFP 下降法也已知对于 R^2 中的二次函数，只用了二步就求得极小点。这就说明 DFP 的变尺度迭代给出的方向

$$P^k=B_k f(x^k) \qquad k=0,1,2,\cdots,n-1 \tag{5-101}$$

可能就是关于二次函数共轭正交的。事实上，我们完全可以证明关于二次函数

$$F(x)=\frac{1}{2}x^TGx+b^Tx+c$$

由式（5-85）给出DFP变尺度递推产生的式（5-101）各方向确具有共轭正交于G的性质，也就是：

$$(P^i)^T G P^j = 0 \qquad i \neq j$$

另外，关于二次函数使用DFP递推式（5-85）时，还恒为正定，也就是说

$$p_k^T q_k > 0$$

对于二次函数是恒成立的。

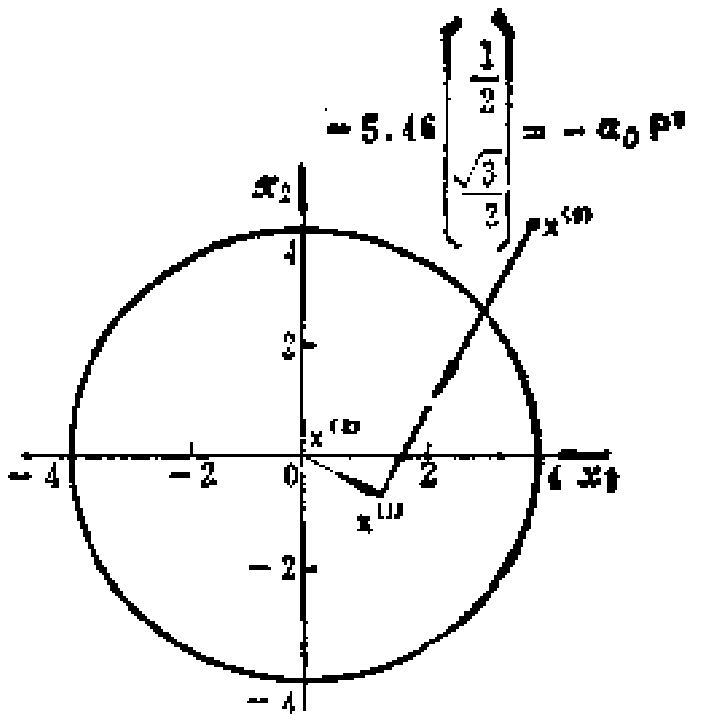


图5-5　二次函数的共轭方向下降方法

仿照关于二次函数的上述共轭方向下降迭代方法，对于一般非二次函数，我们可以用$F'(x)^T$代替二次函数的$Gx+b$，完全套用的给出二种常用的共轭方向法。

首先，我们注意到，对于二次函数

$$F(x) = \frac{1}{2}x^T G x + b^T x + c$$

我们可以采用开始的方向是：

$$P^0 = F'(x^0)^T = \nabla_x F(x^0)$$

然后令
$$P^{k+1} = Gx^{k+1} + b - B_k P^k$$

并使$(P^{k+1})^T G P^k = 0$，就有：

$$B_k = (Gx^{k+1} + b)^T G P^k / (GP^k)^T P^k \tag{5-102}$$

其次，由于：

$$\nabla_x F(x^1) = Gx^1 + b, \quad \nabla_x F(x^0) = Gx^0 + b$$

$$(P^0)^T = \frac{(x^1 - x^0)^T}{\alpha_0} = \frac{[\nabla_x F(x^1) - \nabla_x F(x^0)]^T G^{-1}}{\alpha_0}$$

则给出：

$$[\nabla_x F(x^1) - \nabla_x F(x^0)]^T [\nabla_x F(x^1) - B_0 P^0] = 0$$

同时，因为：

$$\begin{aligned}(Gx^1 + b)^T P^0 &= (Gx^1 + b)^T \Delta_x F(x^0)\\ &= \nabla_x F(x^0)^T (Gx^1 + b)\\ &= \nabla_x F(x^0)^T \nabla_x F(x^1) = 0\end{aligned}$$

所以可知：

$$\mathbf{B}_0 = -\frac{\nabla_x F(\mathbf{x}^1)^T \nabla_x F(\mathbf{x}^1)}{\nabla_x F(\mathbf{x}^0)^T \nabla_x F(\mathbf{x}^0)} \tag{5-103}$$

按式（5-102）和式（5-103）的二种求参数方法，我们就有下面的一般非二次函数共轭方向方法。

（1）丹利尔（Daniel）共轭方向算法：

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{P}^k$$

$$F(\mathbf{x}^{k+1}) = \min\{F(\mathbf{x}^k - \alpha \mathbf{P}^k) | \mathbf{x}^k - \alpha \mathbf{P}^k \in D\}$$

$$\mathbf{P}^0 = F'(\mathbf{x}^0)^T$$

$$\mathbf{P}^{k+1} = F'(\mathbf{x}^{k+1})^T - \mathbf{B}_k \mathbf{P}^k$$

$$\mathbf{B}_k = \frac{F'(\mathbf{x}^{k+1})^T H_F(\mathbf{x}^{k+1}) \mathbf{P}^k}{(\mathbf{P}^k)^T H_F(\mathbf{x}^{k+1}) \mathbf{P}^k}$$

（2）弗勒抽-里弗斯（Fletcher-Reeves）共轭方向算法：

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{P}^k$$

$$F(\mathbf{x}^{k+1}) = \min\{F(\mathbf{x}^k - \alpha \mathbf{P}^k) | \mathbf{x}^k - \alpha \mathbf{P}^k \in D\}$$

$$\mathbf{P}^0 = F'(\mathbf{x}^0)^T$$

$$\mathbf{P}^{k+1} = F'(\mathbf{x}^{k+1})^T - \mathbf{B}_k \mathbf{P}^k$$

$$\mathbf{B}_k = -F'(\mathbf{x}^{k+1}) F'(\mathbf{x}^{k+1})^T / F'(\mathbf{x}^k) F'(\mathbf{x}^k)^T$$

第六章　极小化的直接法

§1　一元函数的极小化

在解析法中，我们要解出无约束目标函数的极小值，总要求出它的梯度函数，有时甚至要求出它的赫森矩阵。如果要避免这两个方面的计算，我们就要采用直接解法，在以后各节，我们将逐步地予以讨论。

问题的另一方面，即使在解析方法中，取定方向的下降迭代，要定出最好的阻尼因子 α 还要进行微分手续，我们在这里也可以改用直接方法求出合于要求的 α 近似值。这样，我们就要讨论直接从数值上极小化

$$\varphi(\alpha)=F(x+\alpha P) \tag{6-1}$$

的问题。

首先，我们来介绍一种用低阶多项式逼近 $\varphi(\alpha)$ 以估计 α^* 的方法。

开始时，先用 a，b 二值试试，如果有 α 是在 a，b 之间而又有：

$$\varphi(a)>\varphi(\alpha),\quad \varphi(\alpha)<\varphi(b)$$

则 $a<\alpha^*<b$。

在 α^* 界定以后，我们可以通过所计算得到的点 $(x^i, F(x^i))$ 做一个低阶多项式函数，极小化这个多项式函数，估计出 $\hat{\alpha}^*$，如果 $\varphi(\hat{\alpha}^*)$ 小于 $\varphi(a)$ 或者小于 $\varphi(b)$，就采用 $\hat{\alpha}^*$；如果不是这样，$\hat{\alpha}^*$ 就用来把区间 (a,b) 缩小为 $(a,\hat{\alpha}^*)$ 或缩小为 $(\hat{\alpha}^*,b)$，然后再在这个小区间进行内插。

具体地说，从 x 在 P 方向的射线上找极小化 $F(x+\alpha P)$ 的

点，可以用α的一个二次函数来逼近。它包括以下几个步序：

（1）计算出$F(x)$和指定α_1的$F(x+\alpha_1 P)$；

（2）如果$F(x)<F(x+\alpha_1 P)$，就计算$F(x-\alpha_1 P)$；

（3）如果$F(x)>F(x+\alpha_1 P)$，就计算$F(x+2\alpha_1 P)$；

（4）把它们表达成三个函数值：

$$F(x+aP),\quad F(x+bP),\quad F(x+cP);$$

（5）近似$F(x+\alpha P)$为：

$$F(x+\alpha P)\approx\beta_0+\beta_1\alpha+\beta_2\alpha^2 \tag{6-2}$$

其中：β_0，β_1，β_2由求解下列三式来决定，即

$$F^a=F(x+aP)=\beta_0+\beta_1 a+\beta_2 a^2$$

$$F^b=F(x+bP)=\beta_0+\beta_1 b+\beta_2 b^2$$

$$F^c=F(x+cP)=\beta_0+\beta_1 c+\beta_2 c^2$$

然后估计$\min\limits_{\alpha} F(x+\alpha P)=F(x+\alpha^* P)$中的$\alpha^*$为：

$$\alpha^*\approx-\frac{\beta_1}{2\beta_2}=-\frac{1}{2}\begin{vmatrix}1 & F^a & a^2\\ 1 & F^b & b^2\\ 1 & F^c & c^2\end{vmatrix}\Bigg/\begin{vmatrix}1 & a & F^a\\ 1 & b & F^b\\ 1 & c & F^c\end{vmatrix}$$

$$=\frac{1}{2}\,\frac{(b^2-c^2)F^a+(c^2-a^2)F^b+(a^2-b^2)F^c}{(b-c)F^a+(c-a)F^b+(a-b)F^c}$$

这里要保持下式成立：

$$\beta_2=\frac{(c-b)F^a+(a-c)F^b+(b-a)F^c}{(a-b)(b-c)(c-a)}>0$$

来保证有极小点的存在。

又因为：

$$F(x)\approx\beta_0,\quad F(x+qP)\approx\beta_0+\beta_1 q+\beta_2 q^2$$

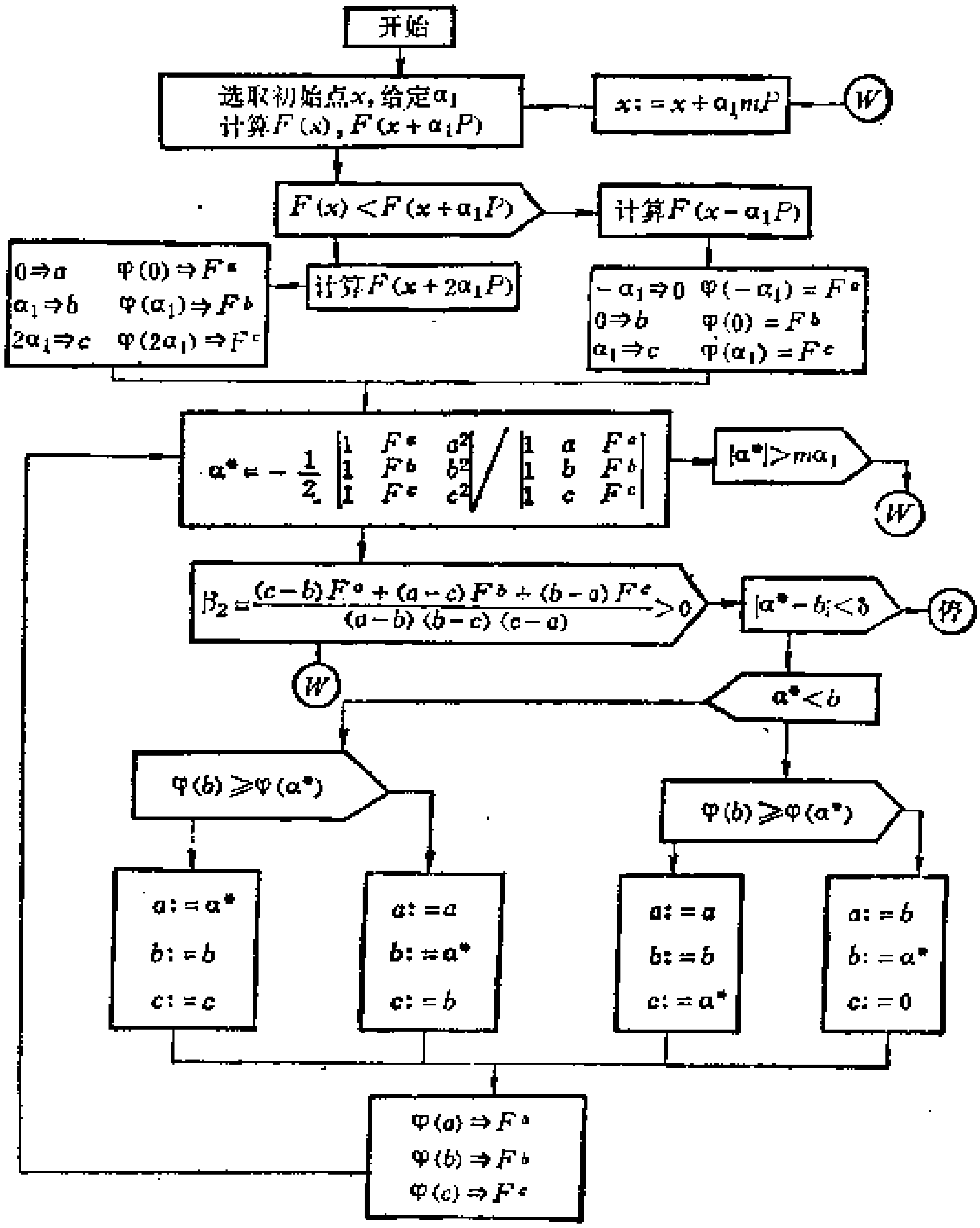
所以α^*又可以近似的表达为：

$$\alpha^*\approx-\frac{\beta_1}{2\beta_2}=\frac{1}{2}\alpha_1-\frac{F(x+\alpha_1 P)-F(x)}{2\alpha_1\beta_2} \tag{6-3}$$

（6）如果$\beta_2<0$或者如果$|\alpha^*|>m\alpha_1$，m是一个定常数（可以容许采取的最大步序数），就在$F(x)$的下降方向中取m步。

（7）采纳α^*的准则是，如果从a，b，c到待预测的极小点$x+\alpha^*P$的一个步长小于一个给定值，就取所预测的极小点是极小。否则，计算$F(x+\alpha^*P)$。

（8）重复地在P方向用二次函数逼近预测极小点，直到极小点落在所需要的精度范围为止。详见下框图：



上述的二次函数逼近求极小点的方法，利用计算机来插点求值已经能够使用了。如果还想尽可能取少一些试点，就可以采用一元函数的优选方法，这就是大家所熟知的一种直接法。

设已知$\varphi(\alpha)$是一个单峰函数，并且已知它在$a_0\leqslant\alpha\leqslant b_0$这个有限区间内有一个极小点，这样一个区间可以经过定h的选取粗略的在a_0，a_0+h，a_0+2h，a_0+4h，……各点计算出$\varphi(\cdot)$直到它开始增值时为止。如果第一步就有：

$$\varphi(a_0)<\varphi(a_0+h)$$

可将h改为$\frac{h}{2}$，再试探计算。这样，就把求$\varphi(\alpha)$极小点的最优解的区间从$(a_0,\ b_0)$缩小到$(a_k,\ b_k)$。假设经过一些步序，还不能确定极小点的位置，我们可以选α_k和α'_k二点是：

$$\alpha_k=\frac{F_{N-1-k}}{F_{N+1-k}}(b_k-a_k)+a_k$$

$$\alpha'_k=\frac{F_{N-k}}{F_{N+1-k}}(b_k-a_k)+a_k \qquad (6-4)$$

$$k=0,1,2,\cdots,N-1$$

其中：N是由我们所要求的精度而选定的整数，同时

$$F_k=F_{k-1}+F_{k-2},\quad F_0=F_1=1 \qquad (6-5)$$

这就是费波纳希（Fibonacci）数列的规律，它们是：

$\{F_n\}$：1，1，2，3，5，8，13，21，34，55，89，…

如果$\varphi(\alpha_k)<\varphi(\alpha'_k)$，则下一个不确定区间选为$(a_{k+1},b_{k+1})=(a_k,\ \alpha'_k)$；如果$\varphi(\alpha_k)>\varphi(\alpha'_k)$，则选$(a_{k+1},b_{k+1})=(\alpha_k,\ b_k)$；如果$\varphi(\alpha_k)=\varphi(\alpha'_k)$，则选$(a_k,\ \alpha_k)$或$(\alpha'_k,\ b_k)$，因为这两个区间的长度是一样的，也就是：

$$b_k-\alpha'_k=\alpha_k-b_k$$

详见图 6-1。

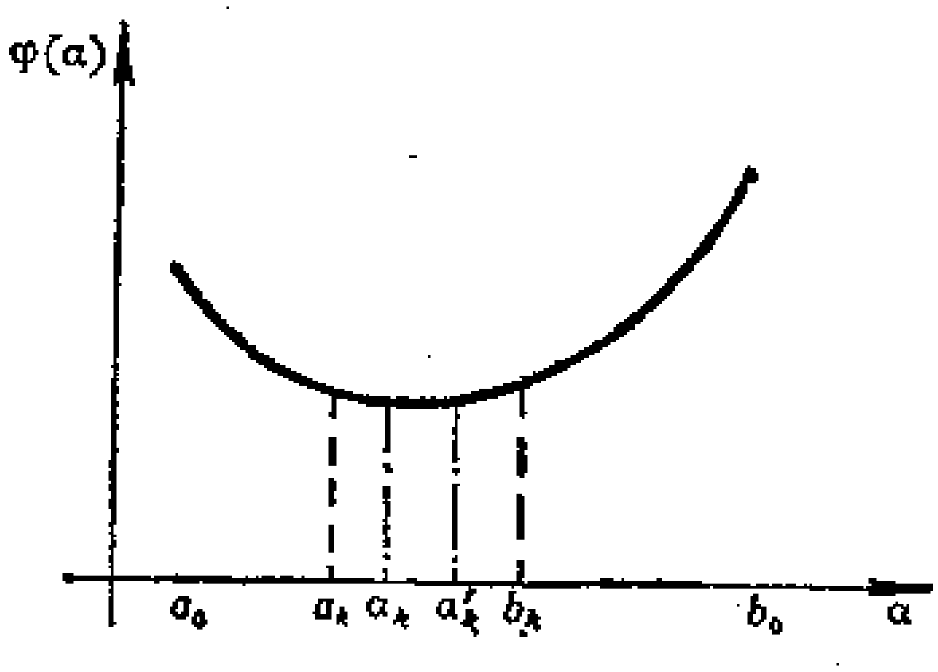


图6-1 费波纳希探索方法

最后的点选定为：

$$\alpha_{N-1}'=\left(\frac{1}{2}+\varepsilon\right)(b_{N-1}-a_{N-1})+a_{N-1}$$

$$\alpha_{N-1}=\frac{1}{2}(b_{N-1}-a_{N-1})+a_{N-1} \qquad (6-6)$$

其中：ε 是一个任意小的正数。这个小正数的引入，目的在于最后决定包含极小点的区间。如果 $\varphi(\alpha_{N-1})<\varphi(\alpha_{N-1}')$，则极小点落在 $(a_{N-1},\ \alpha_{N-1}')$ 中，否则，落在 $(\alpha_{N-1},\ b_{N-1})$，因此最后的不确定区间的长近似的等于 $(b_{N-1}-a_{N-1})/2$。

我们还可以注意到当 $k\to\infty$ 时，则

$$\frac{F_{k-1}}{F_{k+1}}\longrightarrow 0.382$$

$$\frac{F_k}{F_{k+1}}\longrightarrow 0.618$$

所以，式（6-5）的近似公式可改用：

$$\alpha_k=0.328(b_k-a_k)+a_k$$

$$\alpha_k'=0.618(b_k-a_k)+a_k \qquad (6-7)$$

并且

$$\frac{\alpha_k'-a_k}{b_k-a_k}=\frac{b_k-\alpha_k}{b_k-a_k}=0.618$$

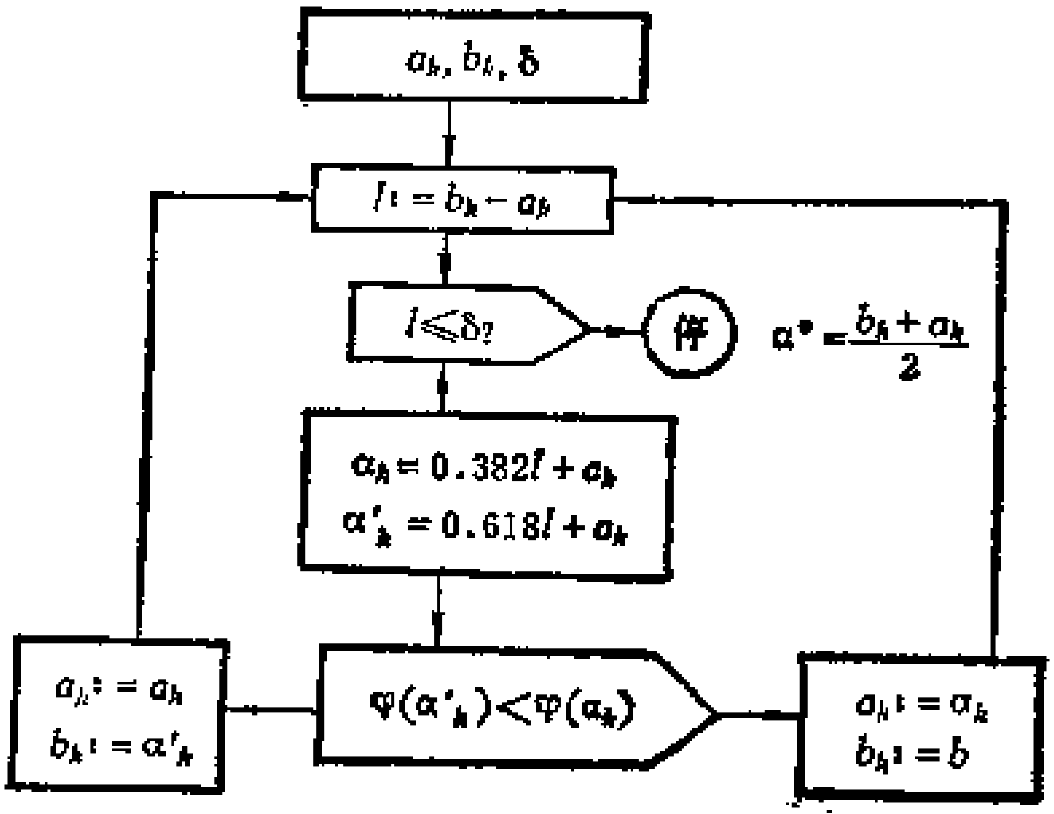
$$b_k-\alpha_k'=\alpha_k-a_k$$

采用式（6-4）或式（6-7）的方法，通常称为“黄金分割法”，原因是 0.618 这个数不仅在费波纳希数列中发生，在古老的黄金分割问题中也产生 0.618 这个数。

关于采用费波纳希数是一种最优选择的数学证明，早在1953年以前，曾经有些统计工作者从统计观点探索最优选法的统计规律●，1953 年才首先由克菲（Kiefer）证明了最优选法的式（6-5）规律●。计算框图如下：

● 见：H. Robbins and S. Monro,《Annals of Math. Statistics》, Vol. 22 p. 1～8 . 1951.
J. Kiefer and J. Wolfowiz,《Annals of Math. Stat.》, Vol. 23 p. 462～, 1952.

● 见：J. Kiefer,《Proceedings of American Math. Soc.》, Vol. 4 p. 502～505, 1953.



事实上，从式（6-5）的差分方程，我们可以令其解是 $r^n = F_n$，则应有下式关系：

$$r^2 = r + 1 \tag{6-8}$$

由此可解得：

$$r = +\frac{1}{2} \pm \frac{\sqrt{5}}{2}$$

于是式（6-5）有解为：

$$F_n = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

而 c_1，c_2 可从初值条件 $F_0 = F_1 = 1$ 来决定。在 n 很大时，第一项为主，故 $F_n \approx c_1\left(\frac{1+\sqrt{5}}{2}\right)^n$，从而有：

$$\frac{F_{n-1}}{F_n} \approx \frac{2}{\sqrt{5}+1} = \frac{\sqrt{5}-1}{2} = \frac{2.2360680-1}{2}$$

$$= \frac{1.2360680}{2} = 0.6180340$$

可见所得值很接近于 0.618。关键是费波纳希数的分点比是：

$$\frac{F_{n-1}}{F_n} \approx \frac{\sqrt{5}-1}{2} \tag{6-9}$$

古老的黄金分割问题是就长为 a 的直线，选定一段长为 x 的分割，见图 6-2，使：

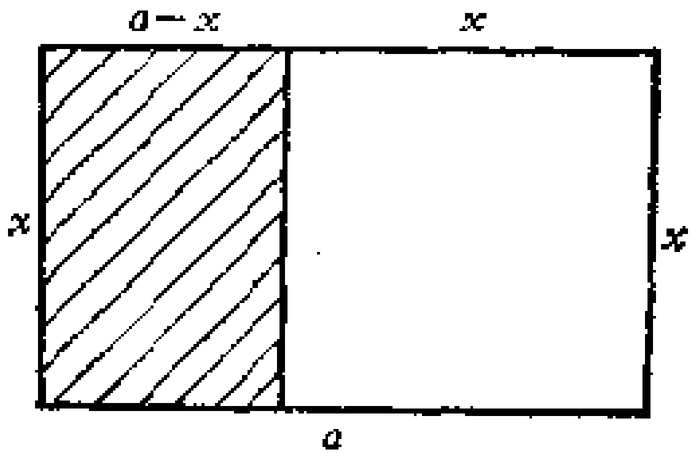
$$a : x = x : (a - x) \quad (6\text{-}10)$$

那么，在求式（6-10）这种关于 x 的外中比问题时，则就有：

$$x^2 + ax - a^2 = 0$$

解出根是：

$$x = -\frac{a}{2} \pm \frac{\sqrt{5}\,a}{2}$$



$a : x = x : (a - x)$

图6-2　黄金分割

取大于零的解，就是：

$$x = \frac{\sqrt{5}-1}{2}a \approx 0.618\,a \tag{6-11}$$

由此可见，它和费波纳希数的比的关系有类似关系。

作图准确求解式（6-11）时，先以 a，$a/2$ 为直角边形成直角三角形 ABC，然后以 C 为心，在 AC 上分 $CD = a/2 = BC$，详见图 6-3，则有：

$$AD = AC - CD = \sqrt{a^2 + \left(\frac{a}{2}\right)^2} - \frac{a}{2} = \frac{\sqrt{5}-1}{2}a$$

有关黄金分割的其它有联系的问题还很多，例如圆内接正十

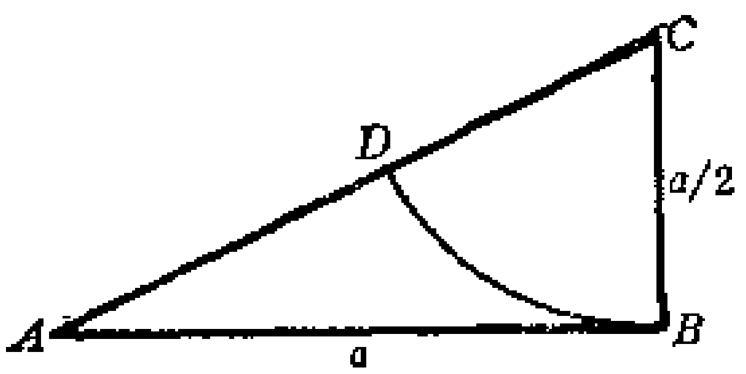


图6-3　黄金分割比的作图

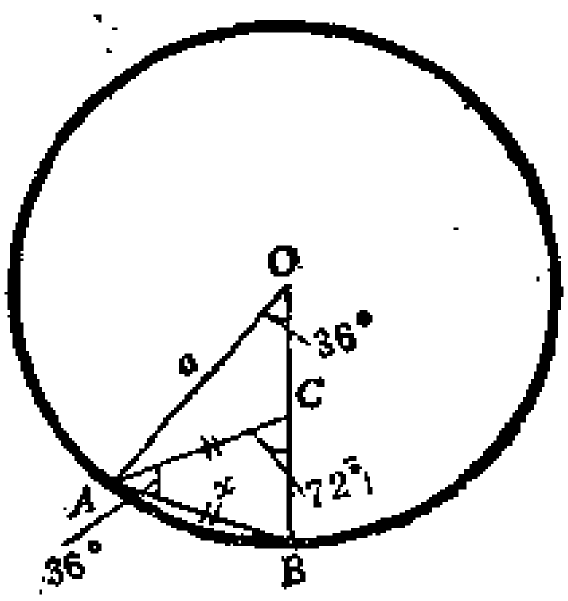


图6-4　圆内接正十边形的边长

边形的一边之长就是圆半径 a 的一个黄金分割，见图 6-4。因为从圆内接正十边形的一边 AB 中的 A 做 AC 交 OB 于 C，且令 $AC=AB$，因

$$\angle ABC=\angle BCA=72^\circ$$

就知道 $\angle CAB=36^\circ$，所以 $\angle OAC=36^\circ$，于是 $AC=OC$。设 $AB=x$，则由于 $\triangle ABC\backsim\triangle ABO$，故 $OA:AB=AB:BC$ 也就是 $a:x=x:(a-x)$。

§2 N.-M. 单形直接法

在完全不用梯度函数和赫森矩阵这些解析方法的直接求多元函数无约束情况下的极小点的方法中，现在最多采用 1964 年涅得和米特（Nelder and Mead）提出的单形直接法●，它是一种容易在数字计算机上实现而行之有效的方法。

所谓“单形”，对于两个变量来说，就是平面上三点形成的等边三角形；对于三个变量来说，就是空间中四点构成的一个正四面形，等等，见图 6-5。

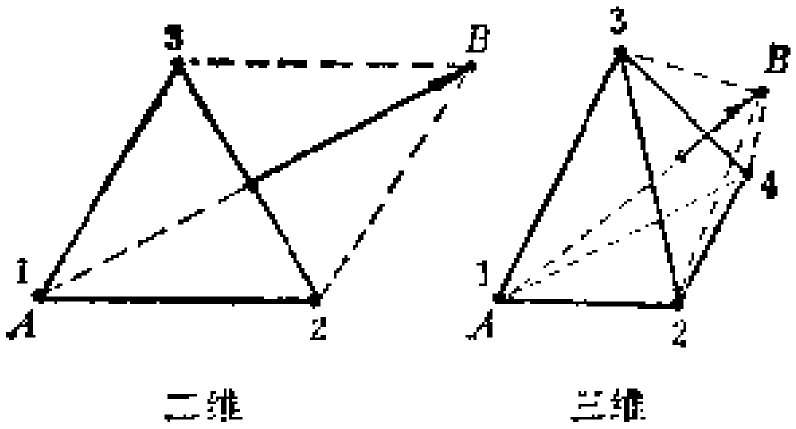


图6-5　二变量和三变量的单形

一般地，在 R^n 空间中，我们可以选定初始点 $\mathbf{x}^0$，这个初始点的选取方法，要看待极小化的目标函数 $F(\mathbf{x})$ 所定义的范围。选定初始点之后，再选一个构成单形的边长 t，找出另外 n 个点使它和 $\mathbf{x}^0$ 一起，由这样的 $n+1$ 个点来做成一个 R^n 中的正规多角形，而以这 $n+1$ 个点做多角形的顶点。

可以证明，按下列方式取 $n+1$ 个点就可以构成 R^n 中的一个边长为 t 的单形：

● 见：A. Nelder and R. Mead,《Computer J.》, 7 p.308~, 1964.

$$\left.\begin{array}{lllll}
\mathbf{x}^0 & \mathbf{x}^{0,1} & \mathbf{x}^{0,2} & \mathbf{x}^{0,3} \cdots & \mathbf{x}^{0,n} \\
x_1^0 & x_1^0+d_1 & x_1^0+d_2 & x_1^0+d_2 \cdots & x_1^0+d_2 \\
x_2^0 & x_2^0+d_2 & x_2^0+d_1 & x_2^0+d_2 \cdots & x_2^0+d_2 \\
x_3^0 & x_3^0+d_2 & x_3^0+d_2 & x_3^0+d_1 \cdots & x_3^0+d_2 \\
\cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\
x_n^0 & x_n^0+d_2 & x_n^0+d_2 & x_n^0+d_2 \cdots & x_n^0+d_1
\end{array}\right\} \qquad (6-12)$$

其中

$$\left.\begin{aligned}
d_1 &= \frac{t}{n\sqrt{2}}(\sqrt{n+1}+n-1) \\
d_2 &= \frac{t}{n\sqrt{2}}(\sqrt{n+1}-1)
\end{aligned}\right. \qquad (6-13)$$

因为这样选点之后，$\mathbf{x}^0$ 和 $\mathbf{x}^{0,i}$($i=1,2,\cdots,n$) 间的距离是：

$$\begin{aligned}
[(n-1)d_2^2+d_1^2]^{1/2} &= \Big\{\frac{t^2}{2n^2}[(n-1)(n+2) \\
&\quad -2(n-1)\sqrt{n+1}+n+1 \\
&\quad +(n-1)^2+2(n-1)\sqrt{n+1}]\Big\}^{1/2} \\
&= \Big\{\frac{t^2}{2n^2}(n^2+n-2+n+1+n^2 \\
&\quad -2n+1)\Big\}^{1/2} = t
\end{aligned}$$

而 $\mathbf{x}^{0,i}$ 和 $\mathbf{x}^{0,j}$($i\neq j$) 间的距离是：

$$[2(d_1-d_2)^2]^{1/2}=\left[2\cdot\left(\frac{t}{\sqrt{2}\,n}n\right)^2\right]^{1/2}=t$$

N.-M. 单形直接法的步序如下：

设 $\mathbf{x}^{k,i}=(x_1^{k,i},\ x_2^{k,i},\ \cdots,\ x_n^{k,i})^T$ ($i=1,2,\cdots,n+1$) 是在 R^n 中第 k 步直接探索的第 i 个顶点($k=0,1,\cdots$)，从开始正规多角形出发（$k=0$ 时），以后单形正规特性逐渐改变。

若将 $\mathbf{x}^{k,i}$($i=1,2,\cdots,n+1$) 各点代入目标函数 $F(\mathbf{x})$ 计值 $F(\mathbf{x}^{k,i})$，并用 $\mathbf{x}^{k,h}$ 和 $\mathbf{x}^{k,l}$ 表示在 $F(\mathbf{x}^{k,i})$ 中达到极大的点

和达到极小的点，即

$$F(\mathbf{x}^{k,h})=\max\{F(\mathbf{x}^{k,1}),\ F(\mathbf{x}^{k,2}),\ \cdots,\ F(\mathbf{x}^{k,n+1})\}$$

$$F(\mathbf{x}^{k,l})=\min\{F(\mathbf{x}^{k,1}),\ F(\mathbf{x}^{k,2}),\ \cdots,\ F(\mathbf{x}^{k,n+1})\}$$

设 $\mathbf{x}^{k,n+2}$ 是除去最坏点 $\mathbf{x}^{k,h}$ 以外，其它各点的中心，也就是这个中心的坐标是：

$$x_j^{k,n+2}=\frac{1}{n}\left[\left(\sum_{i=1}^{n+1}x_j^{k,i}\right)-x_j^{k,h}\right] \tag{6-14}$$

$$j=1,2,\cdots,n$$

然后，按下列四个几何处理，以寻求尽快地找到接近极小点的点。

（1）反射：

将 $\mathbf{x}^{k,h}$ 对中心反射得到一新点为：

$$\mathbf{x}^{k,n+3}=\mathbf{x}^{k,n+2}+\alpha(\mathbf{x}^{k,n+2}-\mathbf{x}^{k,h}) \tag{6-15}$$

其中：$\alpha>0$ 是反射系数，通常取 $\alpha=1$。

（2）扩张：

如果 $$F(\mathbf{x}^{k,n+3})\leqslant F(\mathbf{x}^{k,l})$$

则进行扩张得到一新点为：

$$\mathbf{x}^{k,n+4}=\mathbf{x}^{k,n+2}+r(\mathbf{x}^{k,n+3}-\mathbf{x}^{k,n+2}) \tag{6-16}$$

其中：$r>1$，通常取 $r=2$。

若 $$F(\mathbf{x}^{k,n+4})<F(\mathbf{x}^{k,l})$$

则以 $\mathbf{x}^{k,n+4}$ 代替 $\mathbf{x}^{k,h}$ 构成新的多角形，再选出 $\mathbf{x}^{k+1,h}$，$\mathbf{x}^{k+1,l}$，按上述步骤进行；否则，以 $\mathbf{x}^{k,n+3}$ 代替 $\mathbf{x}^{k,h}$ 进入第 $k+1$ 步。

（3）收缩：

如果 $$F(\mathbf{x}^{k,n+3})>F(\mathbf{x}^{k,l})$$

并且不是对所有 $i\neq h$ 有：

$$F(\mathbf{x}^{k,n+3})>F(\mathbf{x}^{k,i})$$

则仍以 $\mathbf{x}^{k,n+3}$ 代 $\mathbf{x}^{k,h}$。但若对所有 $i\neq h$ 有：

$$F(\mathbf{x}^{k,n+3})>F(\mathbf{x}^{k,i})$$

再看是否

$$F(x^{k,n+3})<F(x^{k,h})$$

如果它成立，就仍以 $x^{k,n+3}$ 代 $x^{k,h}$；或者还有一种情况是：

$$F(x^{k,n+3})>F(x^{k,h})$$

这时就对新、旧 $x^{k,h}$ 进行收缩，选新点为：

$$x^{k,n+5}=x^{k,n+2}+\beta(x^{k,h}-x^{k,n+2}) \tag{6-17}$$

其中：$0<\beta<1$，通常取 $\beta=0.5$。

如果 $$F(x^{k,n+5})<F(x^{k,h})$$

则以 $x^{k,n+5}$ 代 $x^{k,h}$。

（4）缩小：

如果经过收缩之后，仍有：

$$F(x^{k,n+6})>F(x^{k,h})$$

则用

$$\begin{gathered} x^{k,l}+0.5(x^{k,i}-x^{k,l}) \\ i=1,2,\cdots,m \end{gathered} \tag{6-18}$$

代替 $x^{k,i}$ 并计算 $F(x^{k,i})$，进行下一步探索。

探索终止的准则是看是否有：

$$\left\{\frac{1}{n+1}\sum_{i=1}^{n+1}[F(x^{k,i})-F(x^{k,n+2})]^2\right\}^{1/2}\leqslant\varepsilon \tag{6-19}$$

而定。

所以，我们可以将单形直接法的计算过程列成框图见下页。

例 1　用N.-M.单形直接法求

$$F(x)=4(x_1-5)^2+(x_2-6)^2$$

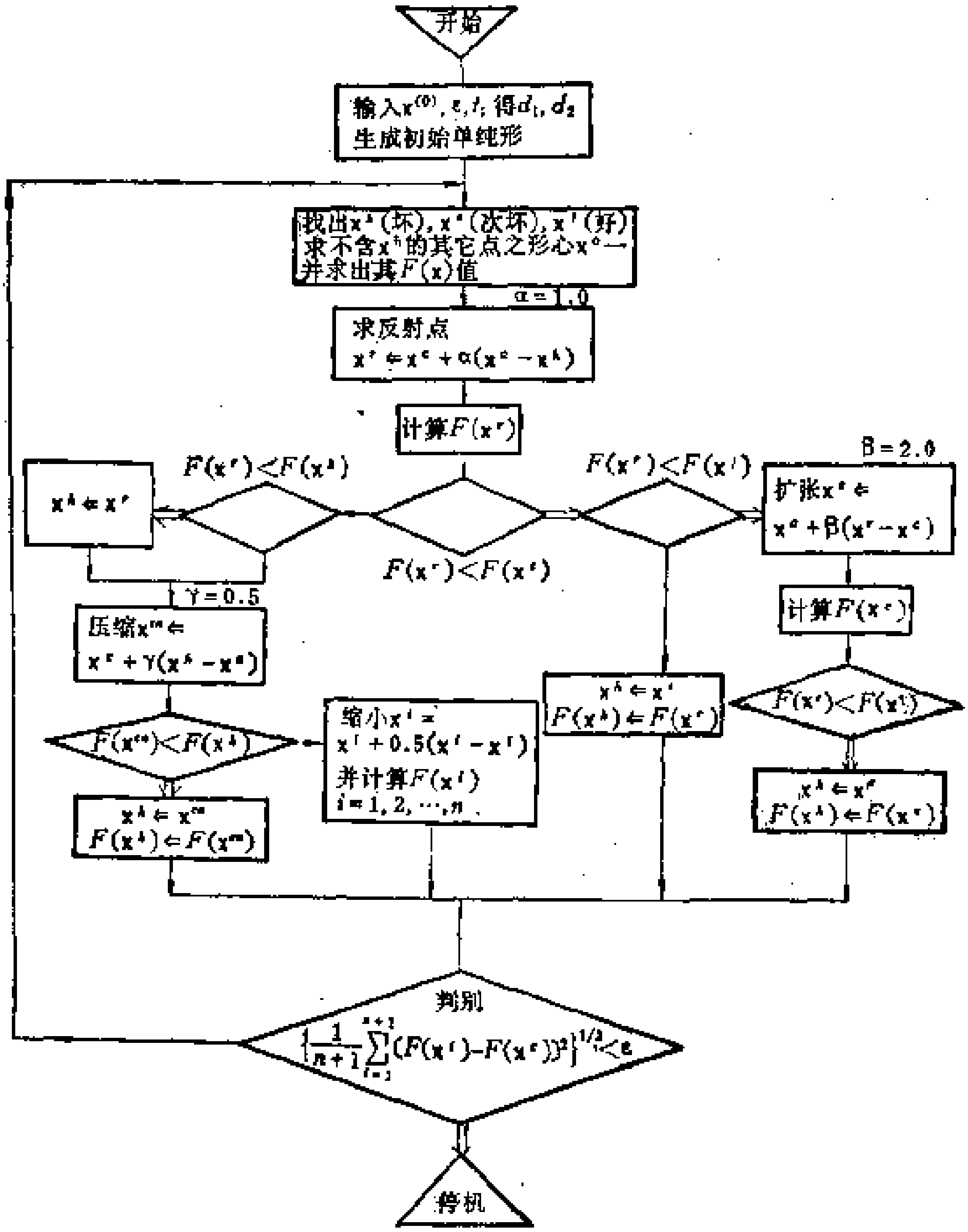
的极小点，已知$x^*=(5,6)^T$。

若开始时取三点是：

$$x^{0,1}=(8,9)^T,\quad x^{0,2}=(10,11)^T \text{ 和 } x^{0,3}=(8,11)^T$$

三点构成不必一定是一个等边形，但总是一个比较简单的三角形。对 $k=0$ 则有：

$$F(8,9)=45,\quad F(10,11)=125,\quad F(8,11)=61$$



所以应将 $x^{0,2}$ 点对 $x^{0,1}$ 和 $x^{0,3}$ 的中心来反射，现在，中心是：

$$x_1^{0,4}=\frac{1}{2}[(8+10+8)-10]=8$$

$$x_2^{0,4}=\frac{1}{2}[(9+11+11)-11]=10$$

反射所得点是：

$$x_1^{0,5}=8+1(8-10)=6$$
$$x_2^{0,5}=10+1(10-11)=9$$

并且

$$F(6,9)=13$$

由于 $$F(6,9)=13<F(8,9)=45$$

故取扩张后的点为：

$$x_1^{0,6}=8+2(6-8)=4$$

$$x_2^{0,6}=10+2(9-10)=8$$

并且

$$F(4,8)=8$$

由于 $F(4,8)=8<F(8,9)=45$，故以 $x^{0,6}$ 代 $x^{0,2}$，并把 $x^{0,6}$ 写成 $x^{1,2}$ 进入下一步序。

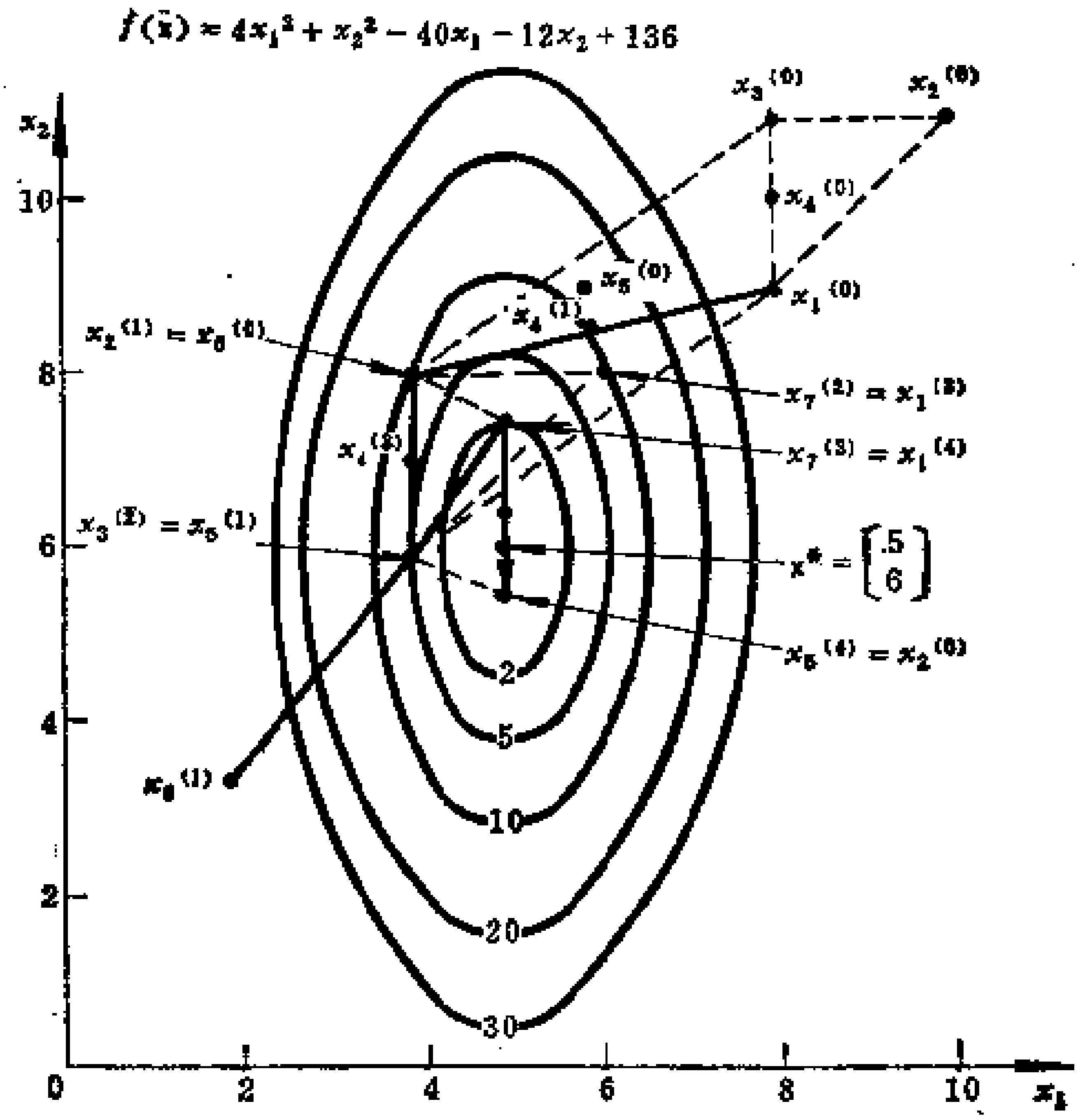


图6-6　N.-M.的无约束极小化直接法

如果我们取 $\varepsilon=10^{-6}$，因为：

$$=\frac{1}{3}[(45-52)^2+(8-52)^2+(65-52)^2]^{1/2}$$

$$=\frac{1}{3}[7^2+44^2+13^2]^{1/2}=26.8>10^{-6}$$

所以，要进入 $k=1$ 的步序。

图 6-6 说明初始步序的探索情况。表 6-1 说明另外四步的顶

表6-1　极小化 $F(\mathbf{x})=4(x_1-5)^2+(x_2-6)^2$ 的 N.-M. 方法的五步计算

步序	多角形顶点	x_1	x_2	$F(\mathbf{x})$ 值	注
0	$\mathbf{x}^{0,1}$	8	9	45	
0	$\mathbf{x}^{0,2}$	10	11	125	$\mathbf{x}^l=\mathbf{x}^{0,1}$
0	$\mathbf{x}^{0,3}$	8	11	65	$\mathbf{x}^h=\mathbf{x}^{0,2}$
0	$\mathbf{x}^{0,5}$	6	9	13	从反射而得
0	$\mathbf{x}^{1,2}=\mathbf{x}^{0,6}$	4	8	8	从扩张而得用 $\mathbf{x}^{0,6}=\mathbf{x}^{1,2}$ 代 $\mathbf{x}^{0,2}$
1	$\mathbf{x}^{2,3}=\mathbf{x}^{1,5}$（反射）	4	6	4	用 $\mathbf{x}^{1,5}=\mathbf{x}^{2,3}$ 代 $\mathbf{x}^{0,3}$
2	$\mathbf{x}^{3,1}=\mathbf{x}^{2,7}$（缩）	6	8	8	用 $\mathbf{x}^{2,7}=\mathbf{x}^{3,1}$ 代 $\mathbf{x}^{0,1}$
3	$\mathbf{x}^{4,1}=\mathbf{x}^{3,7}$（缩）	5	7.5	2.25	用 $\mathbf{x}^{3,7}=\mathbf{x}^{4,1}$ 代 $\mathbf{x}^{3,1}$
4	$\mathbf{x}^{5,2}=\mathbf{x}^{4,5}$	5	5.5	0.25	用 $\mathbf{x}^{4,5}=\mathbf{x}^{5,2}$ 代 $\mathbf{x}^{1,2}$

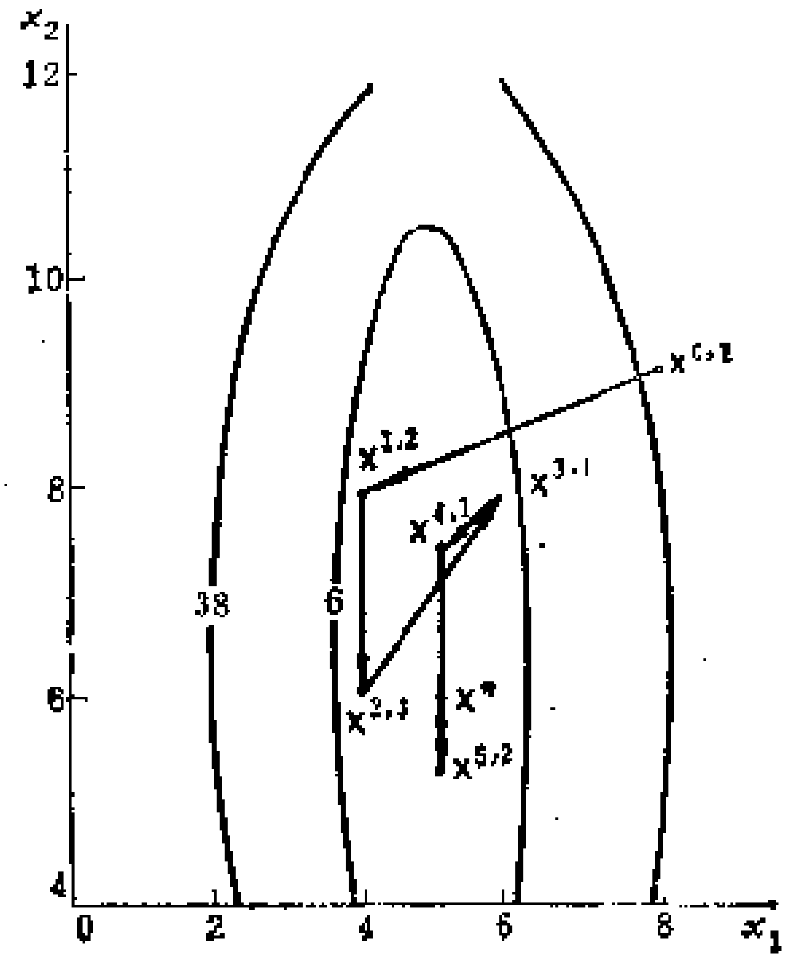


图6-7　N.-M. 方法探索轨线

点和$F(x)$值。图 6-7 表明探索的过程。

§3 P.-H. 的伸缩保差方法

N.-M. 单形直接解法在有约束的情况下，还须要加以改型使用。

1969 年帕威尼和希梅布劳（Paviani-Himmelblau）提出了一种适合于非线性规划的直接解法，采用了一种能够伸缩多角形以保证破坏约束的差值限度，从而给出一种伸缩保差的直接解●。

我们知道，一般非线性规划问题是：

$$
\begin{aligned}
&\text{极小化} \quad F(x),\ x\in R^n \\
&\text{使合于} \quad h_i(x)=0 \qquad i=1,2,\cdots,m \qquad (6\text{-}20)\\
&\qquad\qquad\ g_i(x)\geqslant 0 \qquad i=m+1,\cdots,p
\end{aligned}
$$

严格的要求出容许范围的探索是很费事的，伸缩保差计算使用可行点，同时还采用适当的近可行点。在探索的过程中，逐渐地使近可行点只能向式（6-20）的可行点靠拢。

P.-M. 方法采用一种容许标准是：

$$\Phi^{(k)}=\min\{\Phi^{(k-1)},\Theta^{(k)}\}=\min\left\{\Phi^{(k-1)},\frac{m+1}{r+1}\sum_{i=1}^{r+1}\|x^{k,i}-x^{k,r+2}\|\right\} \tag{6-21}$$

$$\Phi^{(0)}=2(m+1)t$$

其中：t 是初始多角形的边长，如果 x_i 的范围是 (U_i-L_i) 则一般是取：

$$t=\min\left\{\left[\frac{0.2}{n}\sum_{i=1}^{n}(U_i-L_i)\right],\ (U_1-L_1),\cdots,(U_n-L_n)\right\}$$

m 是等式约束的个数，它等于把自由变量的自由度减少到 $n-m=r$；

● 见：D. Paviani and D. M. Himmelblau, 《Operations Res.》, Vol. 17, 1969.

$x^{k,i}$ 是多角形第 i 个顶点（在第 k 步）；

$x^{k,r+2}$ 是我们在 $n+1$ 个顶点中适当选出 $r+1$ 个点，去掉最坏点以后的中心，它反映了 r 个自由度的分散对称部位。

$$\Theta^{(k)}=\frac{m+1}{r+1}\sum_{i=1}^{r+1}\|x^{k,i}-x^{k,r+2}\|$$

$$=\frac{m+1}{r+1}\left\{\sum_{i=1}^{r+1}\sum_{j=1}^{n}(x_j^{k,i}-x_j^{k,r+2})^2\right\}^{1/2} \tag{6-22}$$

它表示了从 $x^{k,i}(i=1,2,\cdots,r+1)$ 到中心 $x^{k,r+2}$ 的平均距离的 $m+1$ 倍，每一约束给出类似的平均距离，总共 m 个约束，至多可以做 m 个代换，所以采用 $\Theta^{(k)}$ 就足以反映点的散开程度了。

因此，现在 $\Phi^{(k)}$ 具有一种正的下降性质，也就是：

$$\Phi^{(0)}\geqslant\Phi^{(1)}\geqslant\Phi^{(2)}\geqslant\cdots\geqslant\Phi^{(k)}\geqslant 0 \tag{6-23}$$

式（6-23）在它的极限情况下，多角形顶点向 $F(x)$ 的约束稳定点收敛，从而使：

$$\lim_{x\to x^*}\Phi^{(k)}=0$$

在迭代使用单形直接法改变多角形时，要每次审定反射点和扩张点是否已经超出可行域，我们要给出每步破坏约束量的度量，定义泛函：

$$T(x)=+\left[\sum_{i=1}^{m}h_i^2(x)+\sum_{i=m+1}^{p}U_i g_i^2(x)\right]^{1/2} \tag{6-24}$$

其中：U_i 是一种算子，当 $g_i(x)\geqslant 0$ 时取 $U_i=0$；而当 $g_i(x)<0$ 时取 $U_i=1$。

如果 $T(x^{(k)})=0$，那么 $x^{(k)}$ 就是一个可行点；在 $T(x^{(k)})\neq 0$ 的情况下，为了使我们的单形直接法能够继续进行，我们并不把所有这些 $x^{(k)}$ 都排斥在外，统认为是完全不可行的点，这时，我们就以容许标准式（6-21）$\Phi^{(k)}$ 为准，若

$$0\leqslant T(x^{(k)})\leqslant\Phi^{(k)} \tag{6-25}$$

则称 $\mathbf{x}^{(k)}$ 是一个"近可行点"；只有在

$$T(\mathbf{x}^{(k)})>\Phi^{(k)} \tag{6-26}$$

时，我们才称 $\mathbf{x}^{(k)}$是"不可行点"。

从而，我们在对有约束极小化问题式（6-20）使用单形直接解法时，我们简化它为：

$$\begin{aligned}&\text{极小化}\quad F(\mathbf{x}),\ \mathbf{x}\in R^n\\&\text{使合于}\quad \Phi^{(k)}-T(\mathbf{x})\geqslant 0\end{aligned} \tag{6-27}$$

在一遇到不可行点时，再用单形法把它拉回到容许的范围之内。直到

$$\Phi^{(k)}\leqslant \varepsilon \tag{6-28}$$

对于给定的充分小正数 $\varepsilon>0$ 成立时为止。也就是，此时由于 $\varepsilon-T(\mathbf{x}^{(k)})\geqslant 0$，当然有：

$$T(\mathbf{x}^{(k)})\leqslant \varepsilon \tag{6-29}$$

例 2　研究非线性规划问题，

$$\text{极小化}\quad F(\mathbf{x})=x_1^2+x_2^2,\ \mathbf{x}\in R^2$$

$$\text{使合于}\quad h_1(\mathbf{x})=x_1^2+x_2^2-9x_2+4.25=0$$

$F(\mathbf{x})$的等值曲线以及使 $h_1(\mathbf{x})=0$ 成立的点的曲线见图 6-8。

问题的解是 $\mathbf{x}^*=(0,0.5)^T$，在 $\mathbf{x}^*$ 点，$F(\mathbf{x}^*)=0.25$，并且 $h_1(\mathbf{x}^*)=0$。根据伸缩保差方法，问题转化为：

$$\text{极小化}\quad F(\mathbf{x})=x_1^2+x_2^2,\ \mathbf{x}\in R^2$$

$$\text{使合于}\quad \Phi^{(k)}-T(\mathbf{x})\geqslant 0$$

其中

$$T(\mathbf{x})=[(x_1^2+x_2^2-9x_2+4.25)^2]^{1/2}$$

注意，这个例子中，$r=n-m=1$，所以用 $r+1=2$ 个顶点来求$F(\mathbf{x})$的极小是可以的，但是现在要用单形法，所以要采用相关点三个点来探索，如果是十个变量，五个等式约束，就只要采用六个顶点来使用单形解法。

我们现在说明二步使用伸缩保差解法处理问题的情况。

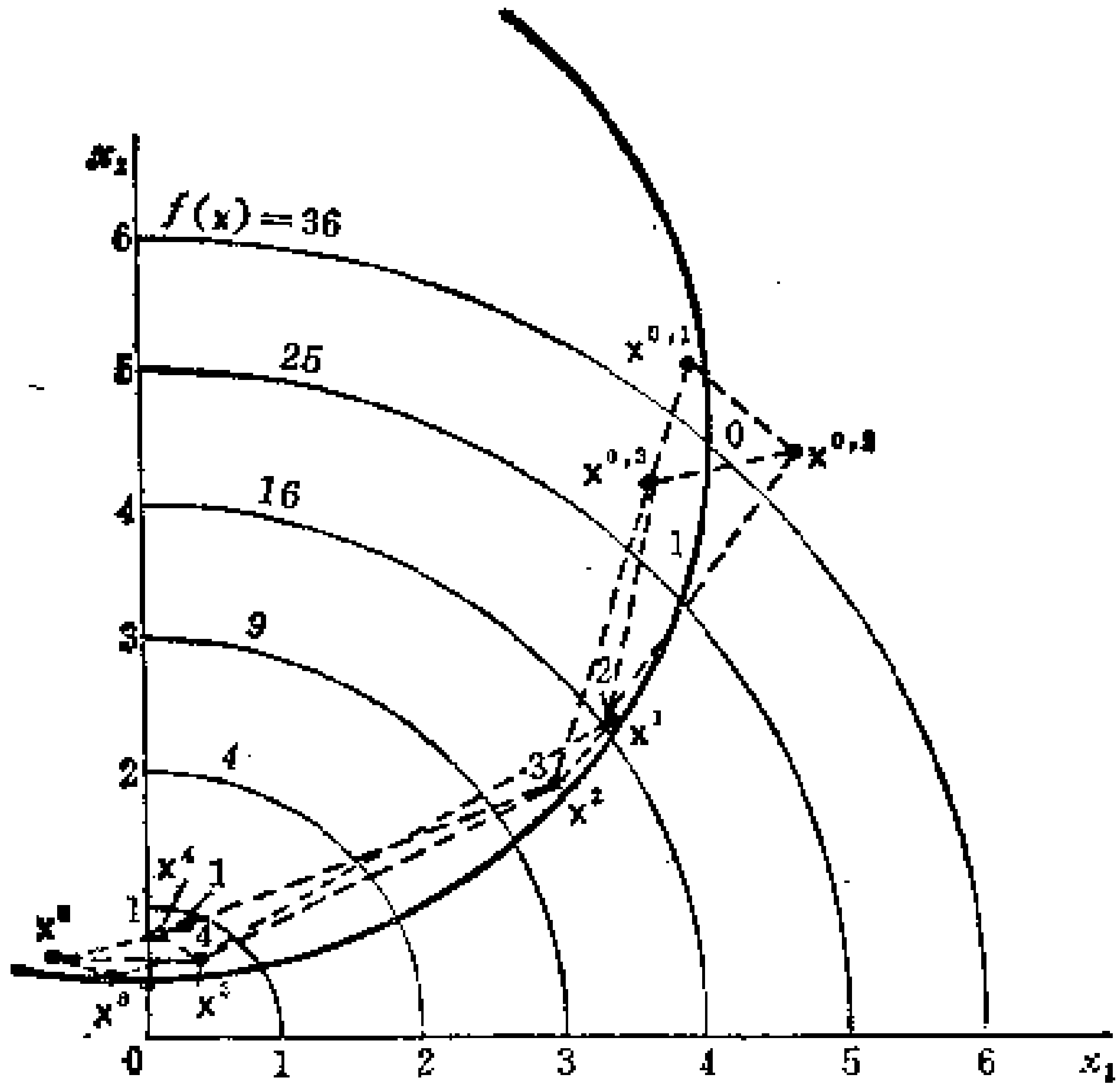


图6-8 伸缩保差方法的有约束极小化

初始点若取 $\mathbf{x}^0=(4, 4.5)^T$，初始多角形的大小是取 $t=1$，那么从式（6-21）可得：

$$t=1,\quad m=1,\quad \Phi^{(0)}=4$$

所以，初始多角形的顶点是：

$$\mathbf{x}^{0,1}=(3.592, 4.092)^T,\quad \mathbf{x}^{0,2}=(4.558,\ 4.351)^T,$$

$$\mathbf{x}^{0,3}=(3.85, 5.06)^T$$

对应的函数值是：

$$F(\mathbf{x}^{0,1})=29.64,\quad F(\mathbf{x}^{0,2})=40.4,\quad F(\mathbf{x}^{0,3})=40.4$$

（Ⅰ）$K=0$ 的探索阶段：

$F(\mathbf{x})$ 最优值的点是 $\mathbf{x}^{0,l}=\mathbf{x}^{0,1}$，这时 $F(\mathbf{x}^{0,l})=29.64$；$F(\mathbf{x})$ 最大值的顶点是 $\mathbf{x}^{0,h}=\mathbf{x}^{0,3}$，这时 $F(\mathbf{x}^{0,h})=40.4$。而在 $\mathbf{x}^{0,l}$ 时 $T(\mathbf{x})$ 表达式为：

$$T(\mathbf{x})=[(x_1^2+x_2^2-9x_2+4.25)^2]^{1/2}$$

其值是 $T(\mathbf{x}^{0,l})=2.65$，而且

$$\Phi^{(0)}-T(\mathbf{x}^{0,l})>0$$

所以 $x^{0,l}$ 是一个近可行点。

在极小化过程中，不必验证其它两个顶点，因为它们还是要用 $F(x)$ 较好值的其它顶点所代替。

排除最坏点外的其它点的中心是：

$$x^{0,4}=(4.07,\ 4.22)^T$$

取 $\alpha=1$，$x^{0,h}$ 对 $x^{0,4}$ 的反射点是 $x^{0,5}=(4.28,\ 3.37)^T$。

计算 $T(x)$ 在 $x^{0,5}$ 的值为 $T(x^{0,5})=3.75$，而现在 $\Phi^{(0)}-T(x^{0,5})>0$，所以 $x^{0,6}$ 是近可行点，并且 $F(x^{0,5})=29.6$，$F(x^{0,5})<F(x^{0,l})$。进一步扩张取 $r=2$，给出 $x^{0,6}=(4.49,\ 2.52)^T$，这时 $T(x^{0,6})=8$。但是 $\Phi^{(0)}-T(x^{0,6})<0$，所以 $x^{0,6}$ 是不可行的；所以，$T(x)$ 必须以 $x^{0,6}$ 开始来极小化，以使

$$\Phi^{(0)}-T(x)>0$$

成立。

经过关于 $x^{0,6}$ 使 $T(x)$ 极小化以后的取代点是 $x^{0,6}=(3.32,\ 2.40)^T$，这时是 $T(x^{0,6})=0.59$。所以 $x^{0,6}$ 是近可行点。

其次，$F(x^{0,6})=16.8$，并且 $F(x^{0,6})<F(x^{0,l})$，所以 $x^{0,h}=x^{0,3}$ 为 $x^{0,6}$ 所取代。$x^{0,6}$ 就是图 6-8 中 x^1，形成新的单形顶点为 $x^{0,1}$，$x^{0,2}$，x^1。完成了 $k=0$ 的步序。

（2）$k=1$ 的探索阶段：

在 $k=1$ 的探索开始时的伸缩多角形的顶点是：

$x^{0,1}=(3.592,\ 4.092)^T$，$x^{0,2}=(4.558, 4.351)^T$，$x^1=(3.32,\ 2.40)^T$ 容许准则 $\Phi^{(1)}=0.745$，而 $F(x)$ 的最大值和最小值点是 $x^{1,h}=x^{0,2}$ 和 $x^{1,l}=x^1$ 并且：

$$F(x^{1,h})=16.8,\quad T(x^{1,l})=0.59$$

由于 $\Phi^{(1)}-T(x^{1,l})>0$，所以探索可以继续进行。

这时，排除最坏点后的中心是：

$$x^{1,4}=(3.45, 3.29)^T$$

而 $x^{1,h}=x^{0,2}$ 对 $x^{1,4}$ 的反射是 $x^{1,5}=(2.34, 2.23)^T$，这时 $T(x^{1,5})=5.50$，而此时 $\Phi^{(1)}-T(x^{1,5})<0$，所以 $x^{1,6}$ 是不可行点。极小

化 $T(\mathrm{x})$，以 $\mathrm{x}^{1,5}$ 为其初始点来进行。

以 $\mathrm{x}^{1,6}$ 为起点极小化 $T(\mathrm{x})$ 以后得到 $\mathrm{x}^{1,5}=(3.1,2.1)^T$，此时 $T(\mathrm{x}^{1,5})=0.64$，从而 $\Phi^{(1)}-T(\mathrm{x}^{1,5})>0$，并且 $F(\mathrm{x}^{1,5})=14.01$。由于 $F(\mathrm{x}^{1,5})<F(\mathrm{x}^{1,l})$，进一步扩张到 $\mathrm{x}^{1,6}=(2.75,\ 0.9)^T$，这时 $T(\mathrm{x}^{1,6})=4.5$。由于 $\mathrm{x}^{1,6}$ 不可行，以 $\mathrm{x}^{1,5}$ 为起点极小化 $T(\mathrm{x})$，取代的 $\mathrm{x}^{1,6}=(2.92,\ 1.90)^T$，这时 $T(\mathrm{x}^{1,6})=0.71$，所以新的 $\mathrm{x}^{1,6}$ 是近可行的。而且 $F(\mathrm{x}^{1,6})<F(\mathrm{x}^{1,l})$，从而 $\mathrm{x}^{1,h}=\mathrm{x}^{0,2}$ 为 $\mathrm{x}^{1,6}$ 所取代。这样就完成了 $k=1$ 的步序。$\mathrm{x}^{1,6}$ 就是图中的 x^2。

如此继续下去。

应该指出的，在图 6-8 中的各点表示的方法是用 $x_1^{(0)}$，$x_2^{(0)}$，$x_3^{(0)}$ 各个表示 $\mathrm{x}^{0,1}$，$\mathrm{x}^{0,2}$，$\mathrm{x}^{0,3}$，同时用 $x^{(1)}$，$x^{(2)}$，$x^{(3)}$，$x^{(4)}$，$x^{(5)}$，$x^{(6)}$ 表示 x^1，x^2，x^3，x^4，x^5，x^6 各点。

第七章　计 算 程 序

§1　P.-H.伸缩保差法FORTRAN程序说明

在第六章中已经介绍了P.-H.伸缩保差方法，关于一般非线性规划问题：

$$
\begin{aligned}
&\text{极小化}\quad F(x),\ x\in D\subset R^n\\
&\text{使合于}\quad \left.\begin{aligned} &h_i(x)=0 && i=1,2,\cdots,m\\ &g_i(x)\geqslant 0 && i=m+1,\cdots,p\end{aligned}\right\}
\end{aligned}
\tag{7-1}
$$

的求解，已知可以通过使用无约束极小化直接方法——N.-M.单纯形法来解决，也就是：

$$
\begin{aligned}
&\text{极小化}\quad F(x),\ x\in D\subset R^n\\
&\text{使单形顶点合于}\quad \Phi^{(k)}-T(x)\geqslant 0
\end{aligned}
\tag{7-2}
$$

其中：$\Phi^{(k)}$是作为第k步时，给定的可变允许"公差"，它是一个下降序列，即

$$\Phi^{(0)}\geqslant\Phi^{(1)}\geqslant\cdots\geqslant\Phi^{(k)}\geqslant 0 \tag{7-3}$$

$T(x)$是标志不满足约束的程度，当$T(x)=0$时，说明所有的约束都被满足，否则就不是全部被满足。但是，一般寻找$T(x)\leqslant\Phi^{(k)}$，(即$\Phi^{(k)}-T(x)\geqslant 0$)的解总是比寻找满足$T(x)=0$的解容易。所以定义：

（1）凡满足$T(x)=0$的解称为可行解；

（2）凡满足$\Phi^{(k)}-T(x)\geqslant 0$的解称为近可行解；

（3）凡满足$\Phi^{(k)}-T(x)<0$的解称为不可行解。

这样，利用N.-M.单纯形法来求解$\min F(x)$，在遇到顶点不满足$\Phi^{(k)}-T(x)\geqslant 0$时，则称这种顶点是不可行点，这时，必

须首先用N.-M.单纯形法来极小化 $T(x)$，使所得到的点拉回到允许范围之内。

因此，用N.-M.单纯形方法来极小化有约束的目标函数 $F(x)$时，只接受可行点或近可行点。这是必须步步经过验证的。一旦所得到的 $x^{(k)}$相对于容许准则 $\Phi^{(k)}$是一个非可行点时，就立即暂停极小化$F(x)$的探索，而转为去寻找另外的 x 点，使它充分靠近可行区域，成为可行点或近可行点。这个过程就是以 $x^{(k)}$作为起始点，来作另外的极小化$T(x)$的探索，以找到满足 $\Phi^{(k)}-T(x)\geqslant 0$ 的点为止。

用N.-M.单纯形法极小化$T(x)$时，首先要在非可行点 $x^{(k)}$的基础上构成一个新的正多角形。为了区别对极小化$F(x)$探索过程中所使用过的多角形点，将对$F(x)$探索使用的多角形点记为 $x_i^{(k)}(i=1,2,\cdots,r)$，其中 $r=n-m$，$r\geqslant 2$；而对极小化$T(x)$所使用的多面体顶点记为 $x_i^{(s)}(i=1,2,\cdots,n)$，其中 s 是表示极小化$T(x)$的步数编号。

$T(x)$极小化过程的起始点 $x^{(0)}$，一般都取 $x^{(k)}$，$x^{(k)}$就是极小化$F(x)$的第 k 阶段探索中所遇到的非可行点。在$T(x)$极小化的过程中，可以得到 $x^{(0)}$，$x^{(1)},\cdots,x^{(s)}$，其中 $x^{(s)}$是使$T(x^{(s)})\leqslant\Phi^{(k)}$成立的顶点。这时，就用 $x^{(s)}$来代替$F(x)$的非可行点 $x^{(k)}$，即 $x^{(s)}\Rightarrow x^{(k)}$。这样就结束了$T(x)$极小化的探索过程，重又回到极小化$F(x)$的探索过程中去●。

本程序的结构是：

主段（MASTER）用于输入，调用子程序FLEXIPLEX，以及控制运算执行的方式。

子程序段（FLEXIPLEX）是该程序用来解决非线性规划问题的主要程序段，所用方法就是伸缩保差法。该子程序段工作时，需调用PROBLEM，SUMR，START，FEASBL，WRITE

● 见，David M. Himmelblau，《Applied Nonlinear Programming》，1972.

等子程序段。

子程序段 (PROBLEM) 反映了所要求的非线性规划实际问题的具体内容，标号 1 下面书写等式约束的内容；标号 2 下面书写不等式约束的内容；标号 3 下面书写目标函数的内容。在求解不同题目时，需要程序员自行编写，具体格式可详见例题。

其它子程序段，如 SUMR，START，WRITE，FEASBL 等可详见各段中的注解 (COMMENT)，它们都是无需改变的标准子程序段。使极小化 $F(x)$ 过程中，所遇到的非可行点靠近可行域的专用子程序段 FEASBL，也是本程序的主要程序段之一。

下面，我们分别就程序中所采用的几个技巧加以说明，并进行讨论。

1．关于子程序 FLEXIPLEX 的主要功能和框图

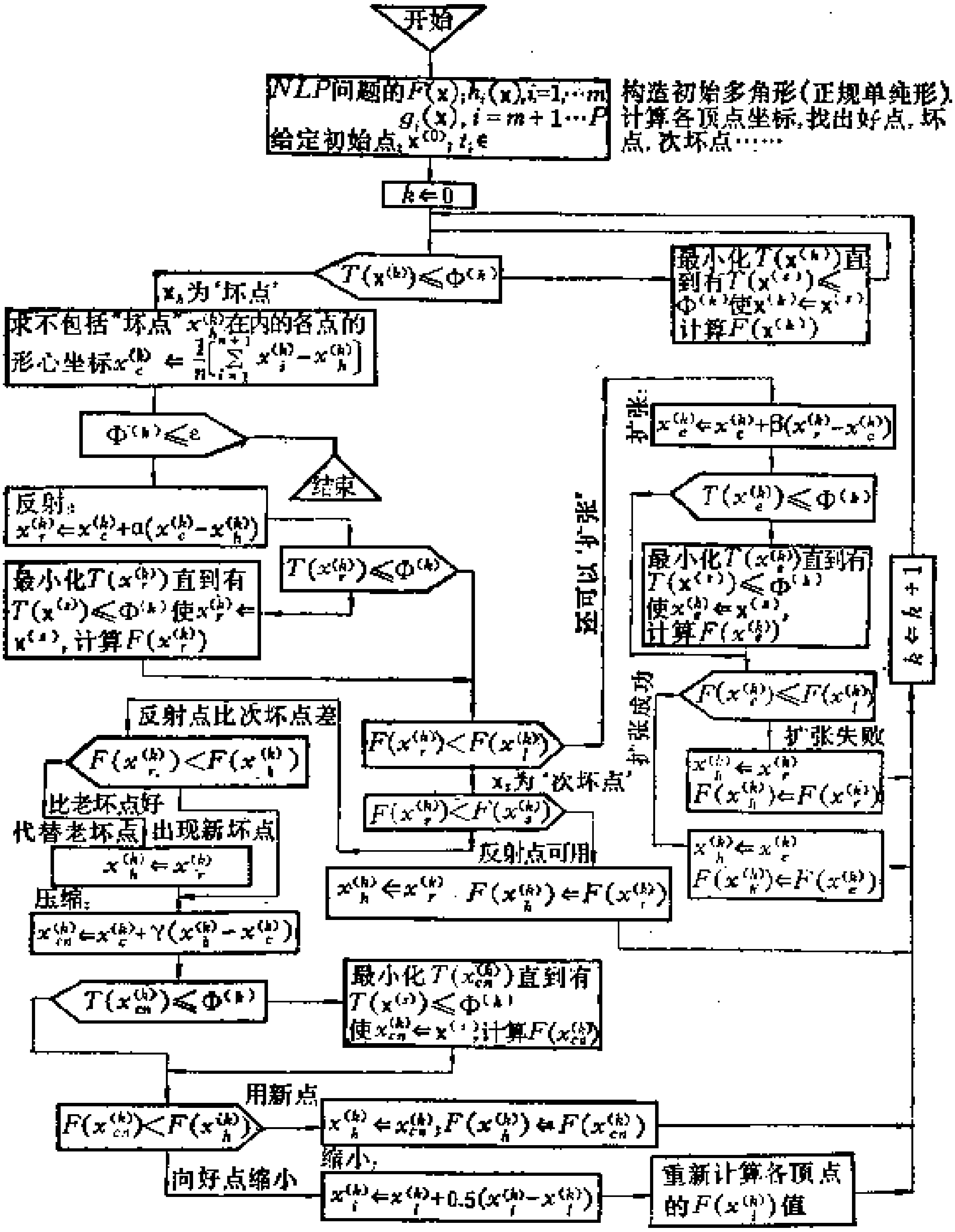
本子程序段是求 $F(x)$ 极小化的专用过程，它的搜索过程所采用的方法就是 N.-M. 单纯形方法，由一个"起始点"作为初始多角形（$n+1$ 个顶点的正单纯形）的形心，并按照给定的边长 t 构成初始多角形，以后就使用"反射"，"扩张"，"收缩"三种搜索方法之一，求得较好的点来代替原单纯形的最坏点。如果所求得点都不比原最坏点好，则向最好点缩小多角形，直到使多角形各顶点到形心平均距离缩小到比预先给定的 ε 还要小时，就结束搜索。应该特别注意的是，每次搜索被采用的点，一定是近可行点或可行点，否则就转到 FEASBL 子程序段，使搜索的新点靠近可行域。总之，处理的过程是很周密的。

本子程序段还用来控制打印计算结果，包括初始信息，中间信息——每 $4(r+1)$ 步输出打印一次，$r=n-m$——和最后信息。

在搜索过程中，经常要计算单纯形顶点，就调用子程序段 START；要计算 $T^2(x)$ 就调用 SUMR 子程序段。另外，本子程序段除了具有各种不同的输出格式语句外，还可调用专门的子程序段 WRITEX 来格式输出目标函数值、各独立变量取值、等式

約束值及不等式约束值。这些子程序段比较简单，容易阅读。

伸缩保差法程序框图如下：



2．关于子程序FEASBL的框图说明

伸缩保差法求解规划问题的关键，就是将有约束的问题转变为无约束的直接探索法，而探索的每一点必须保证它是可行点或近可行点，否则，应该先寻找可行点或近可行点。本子程序段

FEASBL 就是用来寻找可行或邻近可行点的标准子程序，主要步骤如下：

1）设 $x^{(0)}=x^{(k)}$ 是一个非可行点，而 $\Phi^{(k)}$ 是搜索过程在第 k 阶段上由式

$$\Phi^{(k)}=\min\left\{\Phi^{(k-1)},\frac{m+1}{r+1}\sum_{i=1}^{r+1}\|x_i^{(k)}-x_{r+2}^{(k)}\|\right\} \tag{7-4}$$

$$\Phi^{(0)}=2(m+1)t$$

所确定的允许准则值。取 $t=0.05\Phi^{(k)}$ 作为从 $x^{(0)}$ 出发的极小化 $T(x)$ 的初始单纯形的尺度，套用求初始单纯形顶点的子程序段 START，就确定了 $n+1$ 个顶点 $x_i^{(0)}$（$i=1,2,\cdots,n+1$），这些点作为极小化 $T(x)$ 的初始单纯形。然后根据

$$T(x)=\left[\sum_{i=1}^{n}h_i^2(x)+\sum_{i=m+1}^{p}u_ig_i^2(x)\right]^{1/2} \tag{7-5}$$

在这 $n+1$ 个顶点上分别求出 $T(x)$，即

$$T(x_i^{(0)})\qquad i=1,2,\cdots,n+1$$

2）用 $\alpha=1$，$\beta=0.5$，$\gamma=2$ 采用使 $T(x)$ 极小化的 N.-M. 单纯形探索过程，在每阶段的末尾，找出多角形中 $T(x_i^{(r)})$（$i=1,2,\cdots,n+1$）的最小值，并比较 $T(x_i^{(r)})$ 和 $\Phi^{(k)}$ 的值。

3）如果 $T(x_i^{(r)})<\Phi^{(k)}$，说明 $x_i^{(r)}$ 是可行点或近可行点。再若 $T(x_i^{(k)})>0$，就用 $x_i^{(r)}$ 置换不可行点 $x^{(k)}$，以使 $x^{(k)}=x_i^{(r)}$ 被置换为可行点或近可行点。终止 $T(x)$ 的极小化探索过程，转回 FLEXIPLEX 子程序段。

如果 $T(x_i^{(r)})=0$，它很可能是没有等式约束的规划问题（即 $m=0$），这时用N.-M.直接探索 $T(x)$ 的极小，往往效能显著降低，也就是虽然当前是可行点（即 $T(x_i^{(r)})=0<\Phi^{(k)}$），但随后一经反射或扩张，又达到不可行区，即使多次反复探索，只是在可行区和不可行区来回跳动，而并不改善 $T(x)$ 的值。为了减少这种低效能，通常的经验是对仅有不等式约束的问题采用二次内插技术，再转到 7）的处理。

4）如果$T(x_i^{(s)})\geqslant\Phi^{(k)}$，说明$x_i^{(s)}$仍是不可行点，尚须继续返回2）进行探索。但是为了避免单纯形缩成一点，造成死循环，就采用每循环2（$n+1$）次后，检验一次，并计算：

$$A^{(s)}=\frac{1}{n+1}\left[\sum_{i=1}^{n+1}\{T(x_i^{(s)})-T(x_c^{(s)})\}\right]^{1/2}$$

其中：$T(x_c^{(s)})$是第s阶段上单纯形形心上的值。

5）如果$A^{(s)}>10^{-14}$，转回到2），再着手（$s+1$）阶段上的$T(\mathrm{x})$极小化。

6）如果$A^{(s)}\leqslant10^{-14}$，说明伸缩单纯形正要被破坏而成为一个点，但仍未找到可行点或近可行点。这时，N.-M.直接探索过程可能遇到困难，但是我们并不是简单地终止探索，而是设法跳出这个死区，采用的办法是从不可行点$x_i^{(s)}$开始，沿平行于x轴的每一个方向继续搜索$T(\mathrm{x})$的极小值，算法如下：

设x_j^*（$j=1,2,\cdots,n$）是沿平行于x_j坐标轴方向上的一个极小点，它的探索方法是根据$x_i^{(s)}$采用黄金分割方法来实现这种一维搜索。因此，首先在x_1的坐标方向上确定一个包含$T(\mathrm{x})$的极小值的区间l_0，直到使包含x_1^*的l_0尺度减小到比$\Phi^{(k)}$的百分之一更小为止，而后从x_1^*来确定x_2^*，等等。直到所有x_j^*（$j=1,2,\cdots,n$）都被确定。

当然，在每次平行于坐标轴x_j方向的一维搜索结束之前，都要验算$T(x_j^*)\leqslant\Phi^{(k)}$是否成立，如果成立，就确定了新值，以$x_j^*$置换$x_i^{(k)}$，终止$T(\mathrm{x})$的极小化。否则，继续沿下一坐标轴方向搜索，如果所有坐标方向都搜索之后，一个可行点或近可行点仍没有寻找到，则转到1）仍使用N.-M.方法探索（以x_n^*作开始点）。如果这样从1）到6）重复三次仍没有找到可行点或近可行点，则作为失败，打印必要信息，终止极小化过程。

7）当$T(x_i^{(s)})=0<\Phi^{(k)}$，且$m=0$时，采用内点$\mathrm{x}^{(s)}$和$T(\mathrm{x})$极小化中获得的最近外点$\mathrm{x}^{(s-1)}$两点，求得一个中点，用这三个点决定一个二次抛物线来近似函数$T(\mathrm{x})$，这样求得该二次

抛物线的极值点为：

$$x^* = x^{(s)} + \left(\frac{\beta + \sqrt{\beta^2 - 8\alpha z_1}}{4\alpha} \right) \lambda^* \cdot s$$

其中

$$\alpha = z_1 + z_3 - 2z_2$$

$$\beta = 3z_1 - 4z_2 + z_3$$

$$\lambda^* = \left[\sum_{j=1}^{n} (x_j^{(s-1)} - x_j^{(s)})^2 \right]^{1/2}$$

$$s = \frac{x^{(s-1)} - x^{(s)}}{\lambda^*}$$，s 是单位向量。

$$z(x) = \sum_{i=1}^{p} g_i^2(x)$$，p 是 $x^{(s-1)}$ 上违反不等式约束的总数。

$$z_1 = z(x^{(s)})$$

$$z_2 = z(x^{(s)} + 0.5\lambda^* \cdot s)$$

$$z_3 = z(x^{(s-1)})$$

我们就以二次抛物线的极值点 x^* 作为 $T(x)$ 的极小值，这样就得到可行点 x^* 置换 $x^{(k)}$，从而结束 $T(x)$ 的极小化过程。

FEASBL 框图见下页。

下面举例说明：

试求　$F(x) = 4x_1 - x_2^2 - 12$　的极小，使合于

$h_1(x) = 25 - x_1^2 - x_2^2 = 0$

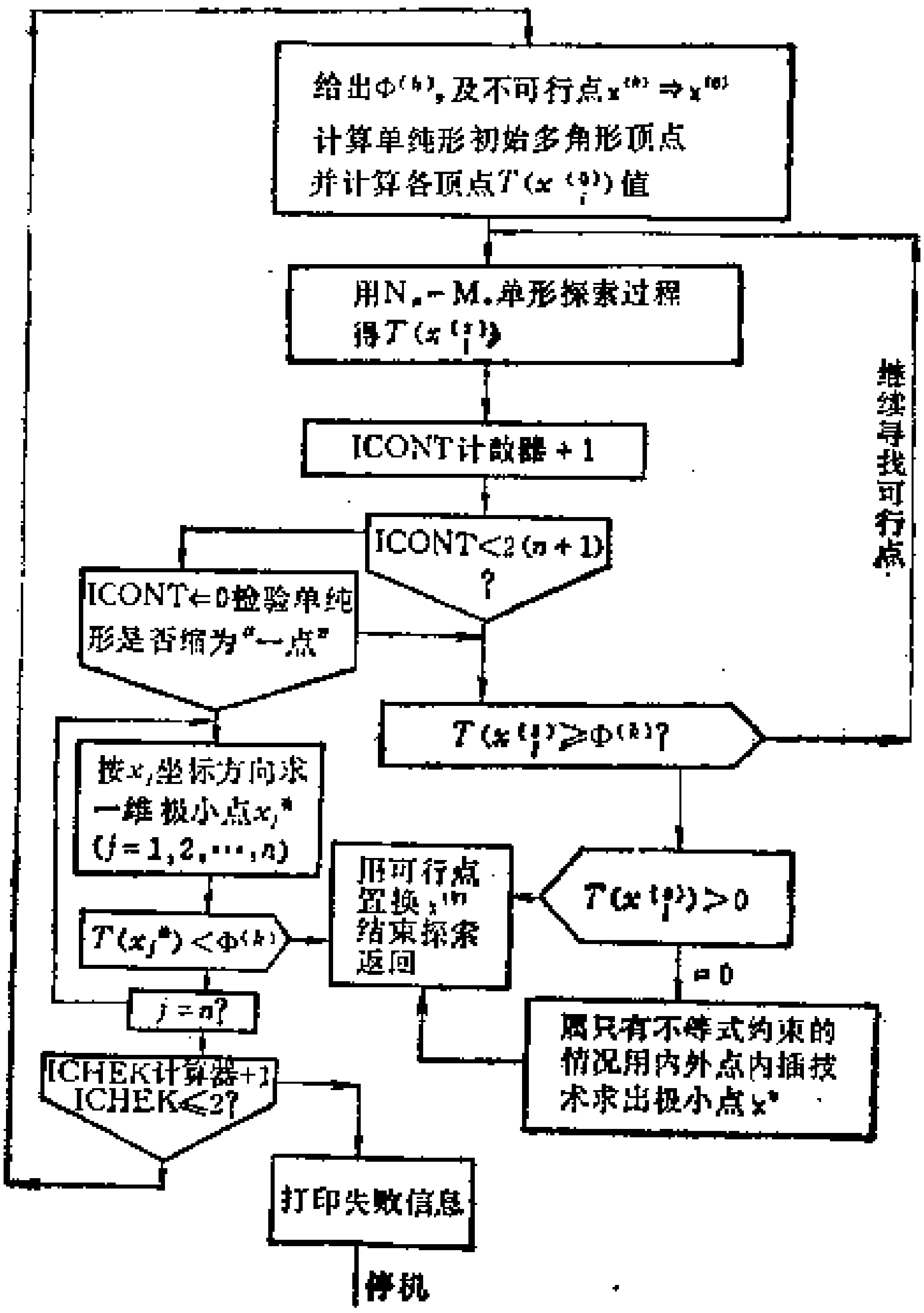
$g_1(x) = 10x_1 - x_1^2 - x_2^2 + 10x_2 - 34 \geqslant 0$

$g_3(x) = x_1 \geqslant 0$

$g_4(x) = x_2 \geqslant 0$

详见图7-1。

对于上述问题，目标函数和约束的计算是通过 SUBROUTINE PROBLEM (INQ) 来解决的，其中参数 INQ 是整型数，取 INQ = 1 时，对应于计算等式约束；取 INQ = 2 时，对应于



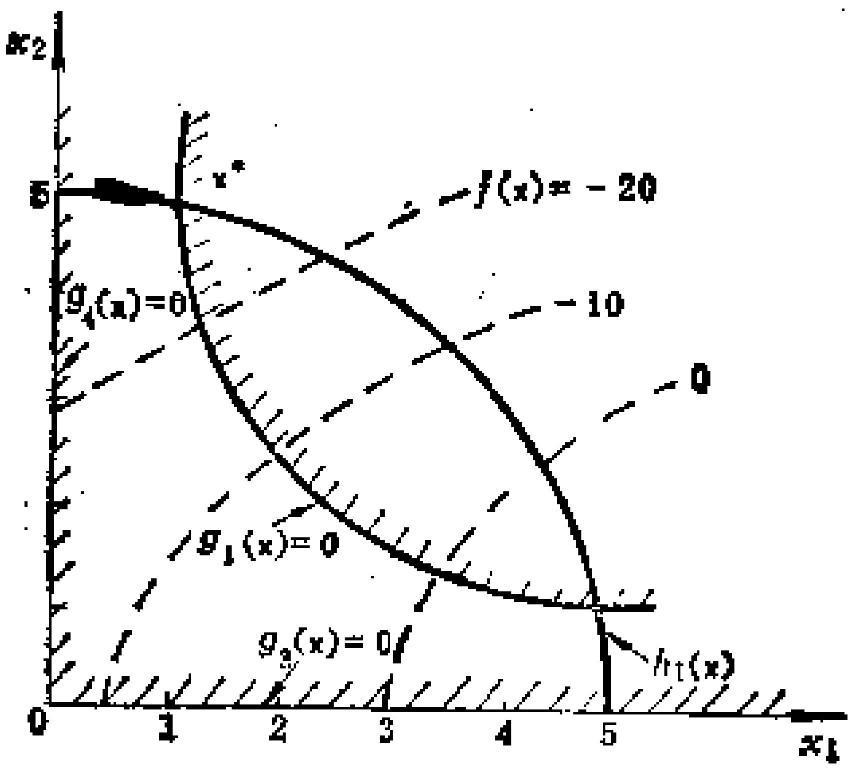


图7-1　求极值点举例图示

计算不等式约束；取 INQ＝3 时，对应于计算目标函数。

我们把等式约束、不等式约束和目标函数个别的式子完全统一地表示为如下的变量形式，即

$$R(I) \qquad I=1, 2, \cdots, m, m+1, \cdots p, \quad p+1$$

我们用上例的 SUBROUTINE PROBLEM(INQ) 子程序段的书写格式来说明使用时，源程序的编写方法是：

```
    SUBROUTINE PROBLEM  (INQ)
    DIMENSION   R(100),    X(50)
    PUBLIC    R,    X
    GOTO    (1, 2, 3)    INQ
C   EQUALITY CONSTRAINTS  (等式约束)
  1   CONTINUE
    R(1)=25.-X(1)*X(1)-X(2)*X(2)
    GOTO    5
C   INEQUALITY CONSTRAINTS (不等式约束)
  2   CONTINUE
    R(2)=10.*X(1)-X(1)*X(1)+10.*X(2)
          -X(2)*X(2)-34.
    R(3)=X(1)
    R(4)=X(2)
C   OBJECTIVE FUNCTION (目标函数)
  3   CONTINUE
    R(5)=4.*X(1)-X(2)*X(2)-12.
  5   RETURN
    END
```

另外，在 MASTER 主段中，尚须对数据输入和表头设计作必要的说明。

主段中读入语句：

READ(1) N, M, P, T, EPSILON 是该例题必须给定

的数据，其对应于设计变量总数，等式约束总数，以及再加上不等式约束的总数的值，t 是单形的边长，ε 就是允许准则收敛值。

READ(1)　(X[I], I=1, N) 是单纯形给定的起始点。

这些数据应由计算人员预先穿好数据纸带，待运算时使用。

本例数据为：

I 2, 1, 4, R0.15, 10^{-5} ⋇⋇⋇<<

R1.0, 1.0, ⋇⋇⋇<<

解题打印的标头，可书写在标号为 11 的格式语句中。如本例书写为：

11　FORMAT (////// 60X, 8HEXAMPLE/60X, ⋇⋇⋇⋇⋇⋇⋇⋇)

其他，在执行运算中须加些什么控制，均可在 MASTER 中进行设计，以利计算方便。

P.-H.伸缩保差法在 DJS-6 机上 FORTRAN 程序见附录一。

§2　解极小化问题的 SUMT 方法　ALGOL60 程序说明

在第三章和第四章，我们已经介绍了序列无约束极小化方法(即 SUMT 方法)。这种方法是通过构造惩罚函数将有约束极小化问题转化为求惩罚函数的无约束极小化问题。为使读者了解这种 SUMT 解法的思路，今将总的解题过程作一介绍。

设 $P(\mathbf{x}, r_k)$ 是所构造的一个惩罚函数，其中 $\mathbf{x}\in R^n$ 是设计变量，r_k 是惩罚因子，或称控制参数。那么，问题的求解过程分为下面两个步骤：

第一步，对固定的 r_k，求惩罚函数 $P(\mathbf{x}, r_k)$ 的无约束极小化问题的解 $\mathbf{x}^*(r_k)$。

第二步，变动 r_k，使 r_k 趋向零，求得问题的解 $\mathbf{x}^*$，即

$\mathbf{x}^*(r_1)$，$\mathbf{x}^*(r_2)$，…，$\mathbf{x}^*(r_k)$，…，趋向 $\mathbf{x}^*$。

关于第一步，固定 r_k 求 $P(\mathbf{x}, r_k)$ 的极小问题，可以用第五

章中牛顿迭代法离散化改型的DFP公式，结合下降法，使用阻尼因子b_k，也就是将公式（5-88）改写为：

$$x^{k+1}=x^k-b_k B_k f(x^k) \qquad k=0,1,2,\cdots \tag{7-6}$$

式中：B_k的取法，用DFP公式（5-85），即

$$B_{k+1}=B_k+\frac{p^k(p^k)^T}{(p^k)^T q^k}-\frac{B_k q^k (q^k)^T B_k}{(q^k)^T B_k q^k} \tag{7-7}$$

b_k的取法是使

$$F(x^{k+1})=F(x^k-b_k B_k f(x^k))<F(x^k)$$

并且使$F(x^{k+1})$尽可能的小。

为了完成上述计算，编写了计算程序●。使用本程序时，可以注意以下各个方面：

1．求解问题的范围

（1）求解问题：

本程序可求解下列三类极小化问题：

1）有约束极小化问题，

$$\begin{aligned} &\text{极小化} \quad F(x), \quad x\in D\subset R^n \\ &\text{使合于} \quad g_i(x)\geqslant 0 \qquad i=1,2,\cdots,m \\ &\qquad\qquad h_j(x)=0 \qquad j=1,2,\cdots,p \end{aligned} \tag{7-8}$$

2）无约束极小化问题，

$$\text{极小化} \quad F(x) \quad x\in D\subset R^n \tag{7-9}$$

3）纯粹等式约束及不等式约束下的解

$$\begin{aligned} &\text{求合于} \quad g_i(x)\geqslant 0 \qquad i=1,2,\cdots,m \\ &\qquad\qquad h_j(x)=0 \qquad j=1,2,\cdots,p \\ &\qquad\qquad x\in D\subset R^n \end{aligned} \tag{7-10}$$

（2）惩罚函数：

本程序采用下列惩罚函数：

● 见：F A. Lootsma, «Boundary Properties of Penalty Functions for Constroined Minimization», Philips Research Reports, 1970, No. 3.

1）求解式（7-8）时用：

$$P_r(x)=F(x)+rb(x)+r^{-1}[l(x)+e(x)] \qquad (7\text{-}11)$$

式中　$b(x)$ 是对数障碍项：

$$b(x)=-\sum_{i\in I_1}\ln g_i(x) \qquad (7\text{-}12)$$

$l(x)$ 与 $e(x)$ 是二次消耗项：

$$l(x)=\sum_{i\in I_2}\{\min[0, g_i(x)]\}^2, \quad e(x)=\sum_{j=1}^{p}h_j^2(x) \qquad (7\text{-}13)$$

其中：I_1，I_2 由下式定义：

$$\begin{aligned} I_1&=\{i|g_i(x_0)>0, \quad 1\leqslant i\leqslant m\} \\ I_2&=\{i|g_i(x_0)\leqslant 0, \quad 1\leqslant i\leqslant m\} \end{aligned} \qquad (7\text{-}14)$$

x_0 是求解问题（7-8）时的初始点。

2）求解式（7-9）时用：

$$P(x)=F(x) \qquad (7\text{-}15)$$

3）求解式（7-10）时用：

$$P(x)=\sum_{i=1}^{m}\{\min[0, g_i(x)]\}^2+\sum_{j=1}^{p}h_j^2(x) \qquad (7\text{-}16)$$

（3）控制参数 r 及收敛准则：

控制参数 r 是正的，递减而趋于零的数列为：

$$\{r_k\}, \quad r_k>0, \quad r_k\to 0$$

程序中初始值 r_0 取为：

$$r_0=\max\left\{10^{-2}, \quad \frac{|V^*|}{100}\right\}$$

式中：V^* 是问题（7-8）中 $F(x)$ 的极小化值的估计值。

r_k 按下式产生：

$$r_k=\frac{r_{k-1}}{\sqrt[3]{10}} \qquad k=1, 2, \cdots \qquad (7\text{-}17)$$

式（7-11）中参数 r 控制着问题（7-8），收敛于最优解 x^*。

设 ε_1 表示给定的相对精度，ε_2 表示绝对精度，$x^{(k,l)}$表示解 x^*的近似值，则当相邻两近似值之差满足

$$|x_j^{(k,l)} - x_{j-1}^{(k-1,l-1)}| < \varepsilon_1 \cdot |x_j^{(k,l)}| + \varepsilon_2 \tag{7-18}$$

$$j = 1, 2, \cdots, n$$

时，计算就立即停止。$x^{(k,l)}$就作为极小的解 x^*的近似值。

2. 总程序结构

总程序的结构如下：

总程序所引入的变量和数组等说明 过程 FUNCTIONS 过程 DERIVATIVES 过程 PRINTTEXT 过程 LINEFEED 主过程 MINIMIZE	}说明部分
总程序语句	}语句部分

(1) 过程 FUNCTIONS

'PROCEDURE' FUNCTIONS (X, GX);

'REAL' 'ARRAY' X, GX; 〈过程体〉;

带有双参数X与GX的过程。它要求对于实型数组X〔1:N〕元素的当前值，能计算出实型数组GX〔1:M〕的元素值。这里X〔1〕，X〔2〕，…X〔N〕是自变数，GX〔1〕，GX〔2〕，…，GX〔M〕是依赖于自变数X〔1〕，X〔2〕，…，X〔N〕的，求解问题的函数，它既包括目标函数$F(x)$，也包括不等式约束和等式约束函数 $g_i(x)$，$h_j(x)$。这个过程要由使用者根据求解课题的要求自行编制。关于如何区别GX〔1〕，GX〔2〕，…，GX〔M〕是目标函数，还是约束函数，这在讨论主过程MINIMIZE时再说明。

例：试编写点集 $A(x_1, x_2, x_3)$ 与点集 $B(x_4, x_5, x_6)$ 之间的最小距离 d 的函数过程，使合于

$$x_1^2+x_2^2+x_3^2\leqslant 5$$
$$(x_4-3)^2+x_5^2\leqslant 1$$
$$4\leqslant x_6\leqslant 8$$

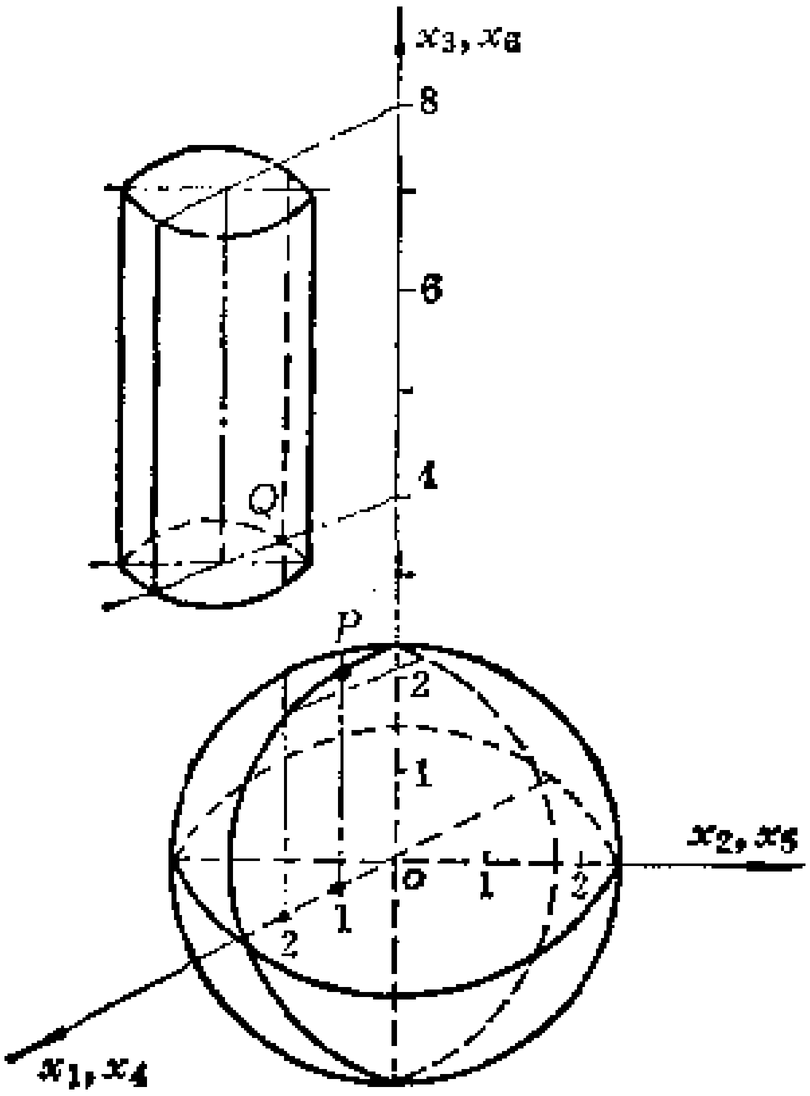


图7-2　举例图解

由图 7-2 可知，式

$$x_1^2+x_2^2+x_3^2\leqslant 5$$

表示以原点为球心，半径为$\sqrt{5}$的球体。所以点集A(x_1, x_2, x_3)在球体上取点。同样，式

$$(x_4-3)^2+x_5^2\leqslant 1,\quad 4\leqslant x_6\leqslant 8$$

表示一个圆柱体，所以点集B(x_4, x_5, x_6)在圆柱体上取点。因此，所提问题就是求这两个几何体间的最短距离问题，可以写成：

极小化　$d^2=g_5(\mathbf{x})=(x_1-x_4)^2+(x_2-x_5)^2+(x_3-x_6)^2$

使合于　$g_1(\mathbf{x})=5-x_1^2-x_2^2-x_3^2\geqslant 0$

$g_2(\mathbf{x})=1-(x_4-3)^2-x_5^2\geqslant 0$

$$g_3(x)=8-x_6\geqslant 0$$

$$g_4(x)=x_6-4\geqslant 0$$

这里自变数有六个，写成X〔1∶6〕，函数有五个，写成GX〔1∶5〕，写成函数过程就是：

```
'PROCEDURE' FUNCTIONS (X, GX);
'REAL' 'ARRAY' X, GX;
'BEGIN'
    GX[1]:=5-X[1]↑2-X[2]↑2-X[3]↑2;
    GX[2]:=1-(X[4]-3)↑2-X[5]↑2;
    GX[3]:=8-X[6];
    GX[4]:=X[6]-4;
    GX[5]:=(X[1]-X[4])↑2+(X[2]-X[5])↑2
              +(X[3]-X[6])↑2
'END';
```

（2）过程 DERIVATIVES

```
'PROCEDURE' DERIVATIVES (X, DGDX);
'REAL' 'ARRAY' X, DGDX; 〈过程体〉;
```

带有双参数X与DGDX的过程，它要求对于实型数组X〔1∶N〕元素的当前值，能计算出实型数组DGDX〔1∶M，1∶N〕元素的值。这里X〔1〕，X〔2〕，…X〔N〕是自变数，DGDX〔I，J〕是函数GX〔I〕关于自变数X〔J〕的一阶偏导数，即

$$DGDX[I,J]=\frac{\partial g_I(x)}{\partial x_J}$$

这个过程也要由使用者根据求解问题的要求自行编制。

例：试编写上段的例中求解问题的导数过程。

由上例可知共有六个自变数（N＝6），五个函数（M＝5），所以

DGDX〔1∶M，1∶N〕＝DGDX〔1∶5，1∶6〕

$$DGDX[I,J]=\frac{\partial g_I(x)}{\partial x_J}$$

$i=1, 2, 3, 4 \qquad j=1, 2, 3, 4, 5, 6$

由 $g_i(\mathbf{x})$ 的定义可得：

$$\frac{\partial g_1(\mathbf{x})}{\partial x_j}=-2x_j, \qquad j=1, 2, 3; \ \frac{\partial g_1(\mathbf{x})}{\partial x_j}=0$$

$$j=4, 5, 6$$

$$\frac{\partial g_2(\mathbf{x})}{\partial x_4}=-2(x_4-3); \ \frac{\partial g_2(\mathbf{x})}{\partial x_5}=-2x_5$$

$$\frac{\partial g_2(\mathbf{x})}{\partial x_j}=0 \qquad j=1, 2, 3, 6$$

$$\frac{\partial g_3(\mathbf{x})}{\partial x_6}=-1; \ \frac{\partial g_3(\mathbf{x})}{\partial x_j}=0 \qquad j=1, 2, 3, 4, 5$$

$$\frac{\partial g_4(\mathbf{x})}{\partial x_6}=1; \ \frac{\partial g_4(\mathbf{x})}{\partial x_j}=0 \qquad j=1, 2, 3, 4, 5$$

$$\frac{\partial g_5(\mathbf{x})}{\partial x_j}=2(x_j-x_{j+3}) \qquad j=1, 2, 3$$

$$\frac{\partial g_5(\mathbf{x})}{\partial x_{j+3}}=-2(x_j-x_{j+3}) \qquad j=1, 2, 3$$

根据上面求导结果，可得下列求导数的过程：

```
'PROCEDURE' DERIVATIVES (X, DGDX);
'REAL' 'ARRAY' X, DGDX;
'BEGIN'
    'INTEGER' J;
    DGDX:=0.0;
    'FOR'J:=1,2,3'DO'DGDX[1,J]:
            =-2*X[J];
    DGDX[2,4]:=-2*(X[4]-3);
    DGDX[2,5]:=-2*X[5];
    DGDX[3,6]:=-1; DGDX[4,6]:=1;
    'FOR'J:=1,2,3'DO'DGDX[5,J]:
            =2*(X[J]-X[J+3]);
    'FOR'J:=1,2,3'DO'DGDX[5,J+3]:
            =-2*(X[J]-X[J+3])
```

```
'END'
```

(3) 过程 PRINTTEXT

```
'PROCEDURE'PRINTTEXT (N1, ST1, N2, ST2,
  N3, ST3, N4, ST4, N5);
'VALUE'N1, N2, N3, N4, N5;
'INTEGER'N1, N2, N3, N4, N5;
'STRING'ST1, ST2, ST3, ST4; 〈过程体〉;
```

该过程调用时，将印出符号 ST_1, ST_2, ST_3, ST_4, 在主过程 MINIMIZE 打印中间结果及最后结果时都要调用这个过程。

(4) 过程 LINEFEED

```
'PROCEDURE'LINEFEED (N);
'VALUE'N; 'INTEGER'N; 〈过程体〉;
```

这个过程也在打印时用，印刷当前一行内容，调整印刷位置，在主过程 MINIMIZE 中被调用。

(5) 主过程 MINIMIZE

本程序绝大部分是主过程内容。它利用序列无约束的极小化技巧，求目标函数合于约束条件的最优化的解，我们在下段详细讨论这些内容。

3. 主过程 MINIMIZE

我们按主过程中形式参数意义，主过程的结构及主过程与其各内层过程的作用三部分进行讨论。

(1) 主过程的形式参数

```
'PROCEDURE'MINIMIZE (X, FUNCTIONS, XTYPE,
YTYPE, ANALYTIC, DERIVATIVES, N, M, TWO,
RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
IMAX)
```

其意义是：

1) X　实型数组 X[1:N]

调用主过程之前，初始点值必须给出，并放在此处。结束时，

存放最优解 x^*。

2）FUNCTIONS　双参数过程 FUNCTIONS (X, GX)

X〔1：N〕存放自变量值，GX〔1：M〕存放函数值，这里GX〔I〕既包括目标函数，也包括约束函数。

3）XTYPE　整型数组 XTYPE〔1：N〕

在调用主过程 MINIMIZE 之前，必须被放入数 0，1 或 2。如果 XTYPE〔J〕=0，则 X〔J〕等于常数，惩罚函数关于 X〔J〕的导数是零；如果 XTYPE〔J〕=1，则 X〔J〕就是通常的自由变量；如果 XTYPE〔J〕=2，则要求 X〔J〕非负，即 X〔J〕≥0。

说明：引入 XYTYE 的目的是对自变量 X〔J〕进行分类，而对 X〔J〕分类的目的，是为了使程序更具有通用性，对于自变数 X〔J〕，有些可作为通常的自变数，有些也可看作是约束，如 X〔J〕等于常数或非负值等。当然，这些约束，我们也可不考虑对自变量的限制，而写成函数形式。

4）GTYPE　整型数组 GTYPE〔1：M〕

在调用主过程之前，必须被放入 0，1，2 或 3。如果 GTYPE〔I〕=0，则 GX〔I〕自动的从课题中排除；如果 GTYPE〔I〕=1，则 GX〔I〕表示目标函数；如果 GTYPE〔I〕=2，则 GX〔I〕表示不等式约束 GX〔I〕≥0；如果 GTYPE〔I〕=3，则 GX〔I〕表示等式约束 GX〔I〕=0。

说明：程序中把目标函数 $F(x)$、不等式约束 $g_i(x)$ 及等式约束 $h_j(x)$ 统一写成 GX〔1：M〕，而由 GTYPE〔1：M〕中预先放入不同值 0，1，2 或 3 来区别 $g_i(x)$ 属于哪一类，从而对 $g_i(x)$ 进行不同的运算。

5）DERIVATIVES　双参数过程 DERIVATIVES (X, DGDX)

X〔1：N〕是自变数，DGDX〔1：M，1：N〕存放导数值，即

$$\text{DGDX〔I，J〕}=\frac{\partial g_i(x)}{\partial x_j}\qquad i=1,2,3,\cdots,m,\qquad j=1,2,\cdots,n$$

6）ANALYTIC　布尔变量

如果ANALYTIC取'TRUE'，那么，按求导过程DERIVATIVES计算课题函数的导数。

如果ANALYTIC取'FALSE'，那么，程序自动按差商法求导。

7）N　整型量

表示数组X[1:N]与XTYPE[1:N]的维数。

8）M　整型量

表示数组GTYPE[1:M]的维数。

9）TWO　尔布变量

如果TWO取'TRUE'，用差分法求导时，每求一次给出两个函数值；否则，给出四个函数值。

10）RAXMIN　实型量

解的相对精度，一般取 10^{-5}。

11）AAXMIN　实型量

解的绝对精度，一般取 10^{-5}。

12）ESTIMATE　实型量

目标函数的最优解的估计值，用以计算控制参数 r_0。

13）CONVERGED　布尔变量

如果满足过程MINIMIZE规定的收敛准则就取'TRUE'；否则取'FALSE'。

说明：这种取'FALSE'的情况是可能发生的，例如：运算迭代的次数超过使用者给定的允许次数，或者约束极小化求解中所用到的无约束极小化的循环次数超过预定的次数。

14）COC　整型量

条件输出控制参数。在调用主过程之前必须放入0，1或2。

当COC=0时，表示不需输出；

当COC=1时，每次迭代有一个短输出，在无约束极小化循环中，只有在开始及最后一次迭代中有：X——输出和G——输出；

当COC＝2时，表示是X——输出。

有关短输出，X——输出，G——输出的内容，到讨论主过程的输出时再介绍。

15）IMAX　整型量

由使用者给定的最大允许迭代次数，当IMAX＝0时，则迭代次数不受限制。

（2）主过程MINIMIZE的结构

- 主过程 MINIMIZE
  - 过程RCYCLE
    - 过程PENALTY
    - 过程GRADIENT
    - 过程LINEMIN
      - 过程READY
      - 过程SHIFT 1
    - 过程OUTPUT 1
  - 过程EXTRAPOL
  - 过程OUTPUT 2
  - 过程OUTPUT 3
  - 过程VECVEC
  - 过程MATVEC

1）函数过程PENALTY

'REAL''PROCEDURE' PENALTY (P, T, Q, GT, REJECT)

（a）形式参数意义：

a）P　实型数组　P〔1：N〕

b）T　实型量

c）Q　实型数组Q〔1：N〕

说明：本程序是计算惩罚函数在点XT〔1：N〕＝P〔1：N〕＋T＊Q〔1：N〕处的值，这里P〔1：N〕，T，Q〔1：N〕构成计算点XT〔1：N〕的位置，P表示初始点，Q表示下降方向，T是加快收敛的因子。

d) GT 实型数组 GT[1:N]

存放函数值（包括目标函数及约束函数）。

e) REJECT 整型量

当 REJECT = 0 时，计算点处于可行域；

当 REJECT > 0 时，计算点处于非可行域。

(b) 功能：

计算惩罚函数 $P_r(x)$ 在点 P[1:N]+T*Q[1:N] 处的值。计算公式是：

$$P_r(x)=F(x)-r\sum_{i\in I_1}\ln g_i(x)+r^{-1}\left\{\sum_{i\in I_2}[\min(0,\ g_i(x))]^2+\sum_{j=1}^{p}h_j^2(x)\right\}$$

这里 I_1，I_2 由（7-14）定义。

在此程序中用

PENALTY=PEN-R*LNBARRIER+LOSS/R 表示。其中：PEN 是目标函数；LNBARRIER 是障碍项，LOSS 是消耗项。

(c) 示意框图（见下页）。

2) 过程 GRADIENT

'PROCEDURE'GRADIENT (XT, GT, DPTDX, ANALYTIC)

(a) 形式参数意义：

a) XT 实型数组 XT[1:N]

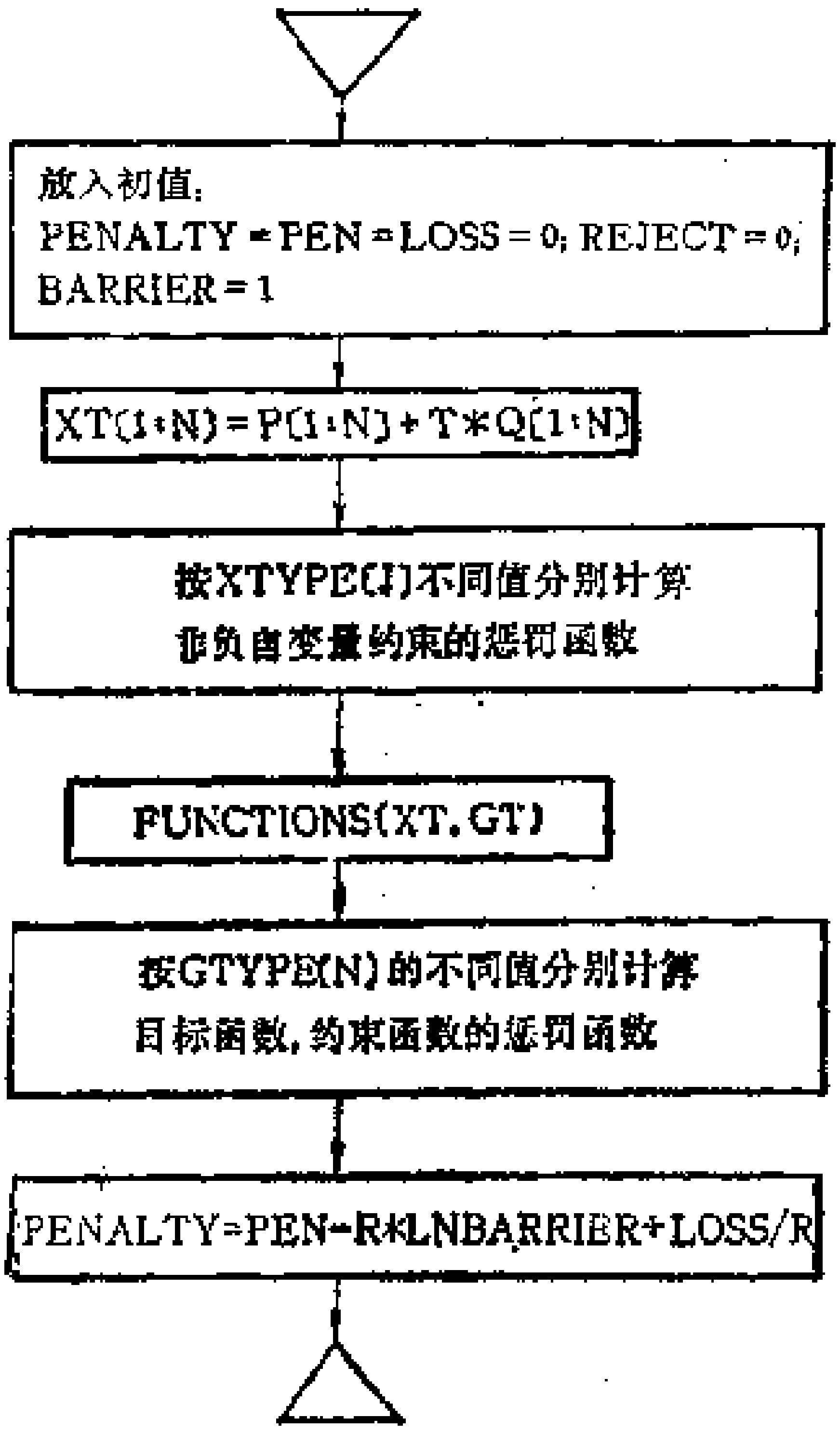
计算梯度的点。

b) GT 实型数组 GT[1:M]

问题函数在点 XT[1:N]处的值。

c) DPTDX 实型数组 DPTDX[1:N]

惩罚函数在点 XT[1:N]处的梯度值。



d) ANALYTIC　布尔变量

ANALYTIC 取'TRUE'用求导公式计算梯度,否则,用差商法计算梯度。

(b) 功能:

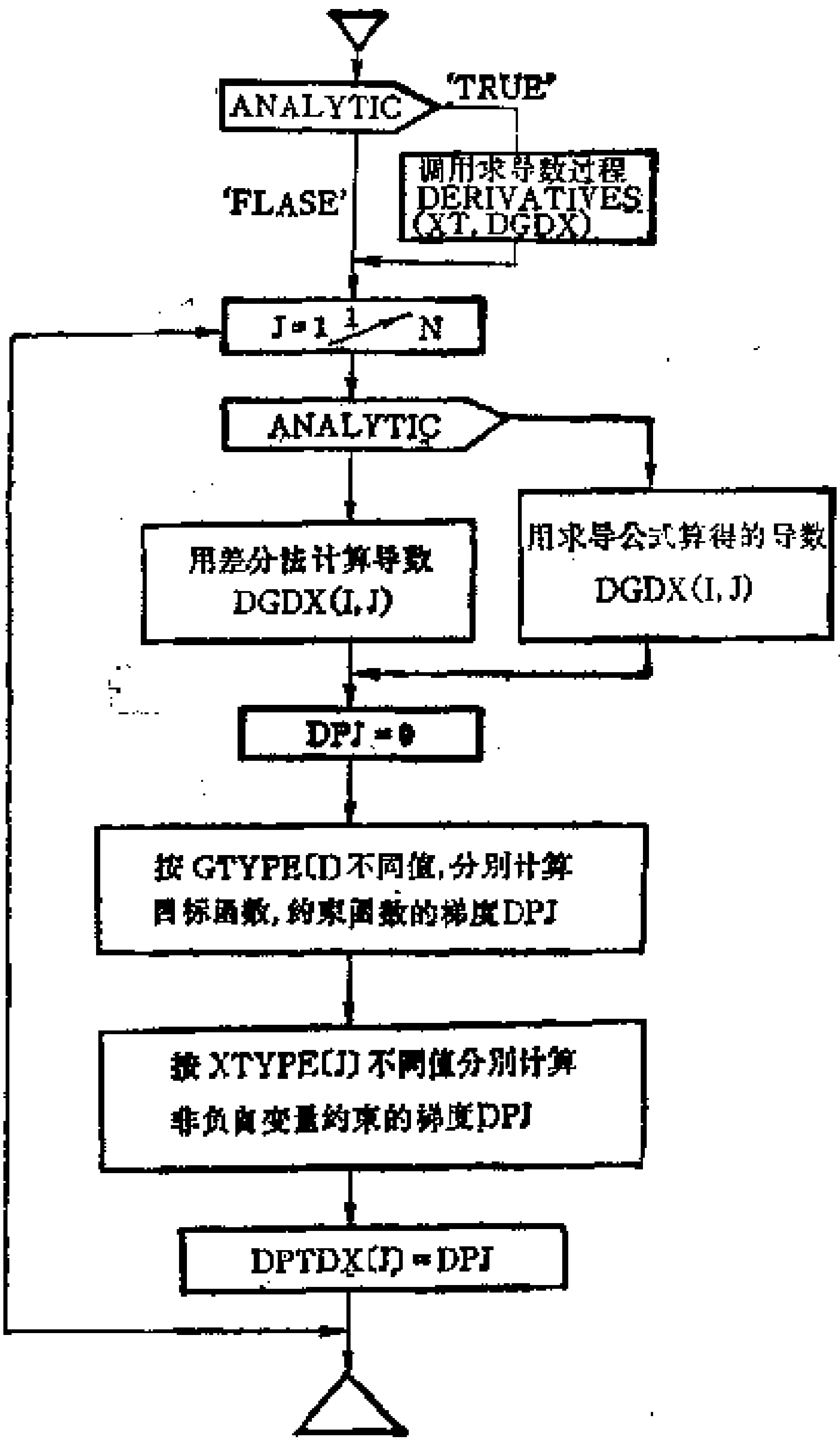
计算惩罚函数$P_r(x)$ 在点 XT〔1:N〕处的梯度,计算公式为:

$$\text{DPTDX}[J]=\frac{\partial P_r(x)}{\partial x_j}=\frac{\partial}{\partial x_j}[\text{PEN}-\text{R}*\text{LNBARRIER}+\text{LOSS}/\text{R}]$$

$$P_r(x)=F(x)-r\sum_{i\in I_1}\ln g_i(x)+\frac{1}{r}\left\{\sum_{i\in I_2}[\min(0,\ g_i(x))]^2+\sum_{j=1}^{p}h_j^2(x)\right\}$$

由于问题函数包含目标函数、约束函数，因此，要根据不同情况代入求导公式。

（c）示意框图：



3）过程 READY 与 SHIFT 1

布尔函数过程 READY（NEW，OLD）：

当 READY 取值'TRUE'时，表示沿方向 S〔1：N〕线性搜

索中，两点已逼近预定的收敛精度；反之，表示未满足收敛精度。

过程 SHIFT (Y, Z, PRY, PRZ, GY, GZ)

分别以点 Z，点 Z 的惩罚函数值 PRZ 及点 Z 处课题函数值 GZ，取代对应的 Y，PRY 及 GY 的值，即

Y⇐Z；PRY⇐PRZ；GY⇐GZ。

4）过程 LINEMIN

'PROCEDURE' LINEMIN (POINT, PRPOINT, GPOINT, GRADPR, S, DIFFP, DIFFG, RACC, AACC)

（a）形式参数意义：

a）POINT　实型数组 POINT[1:N]

存放搜索的开始点及最后点的值。

b）PRPOINT　实型量

惩罚函数在点 POINT 处的值。

c）GPOINT　实型数组 GPOINT[1:M]

问题函数在点 POINT 处的值。

d）GRADPR　实型数组 GRADPR[1:N]

惩罚函数在点 POINT 处的梯度值。

e）S　实型数组 S[1:N]

线性搜索方向。

f）DIFFP　实型数组 DIFFP[1:N]

线性搜索的最后点与出发点之间差值。

g）DIFFG　实型数组 DIFFG[1:N]

线性搜索的最后点梯度与出发点梯度之间的差值。

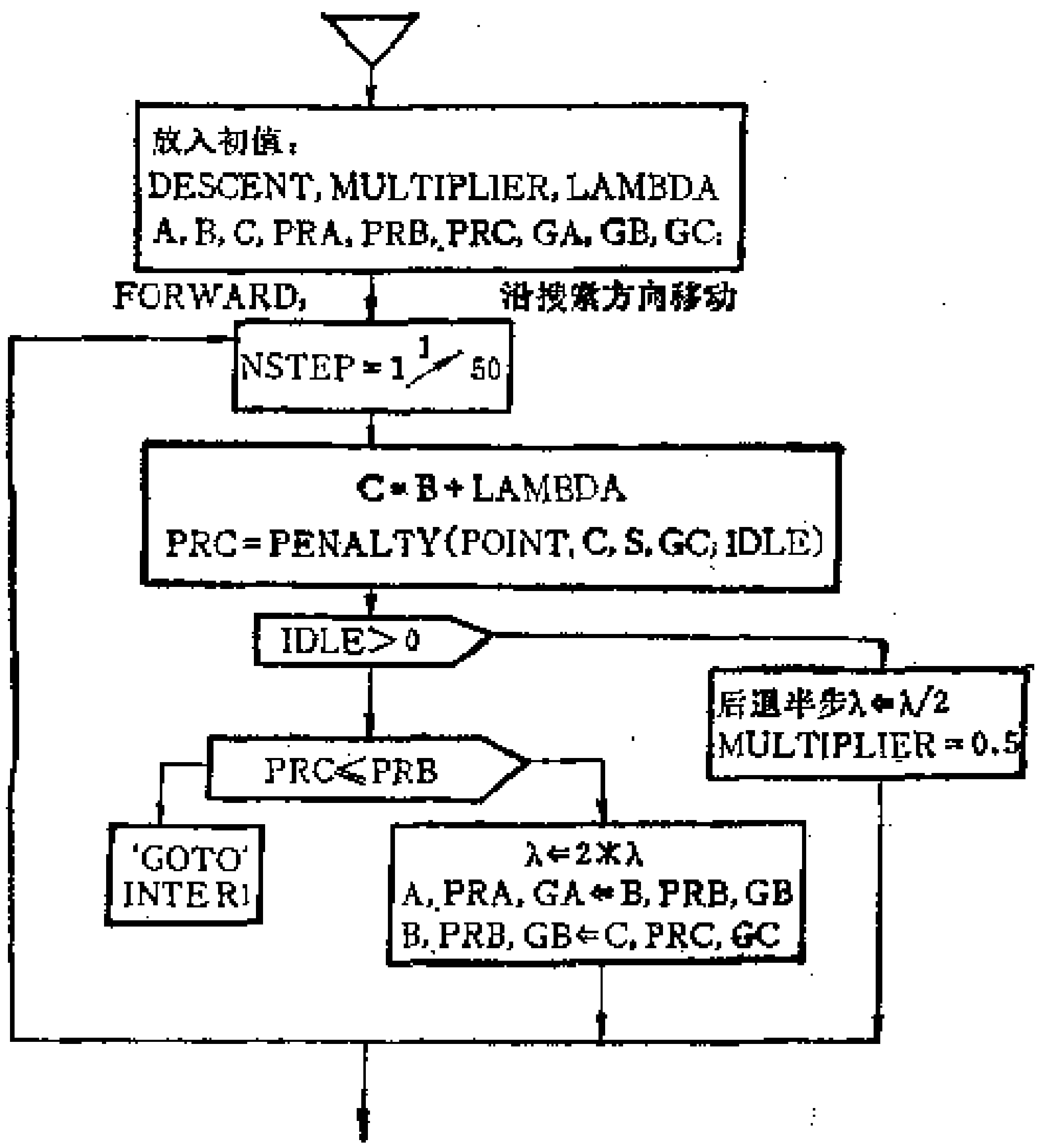
（b）功能：

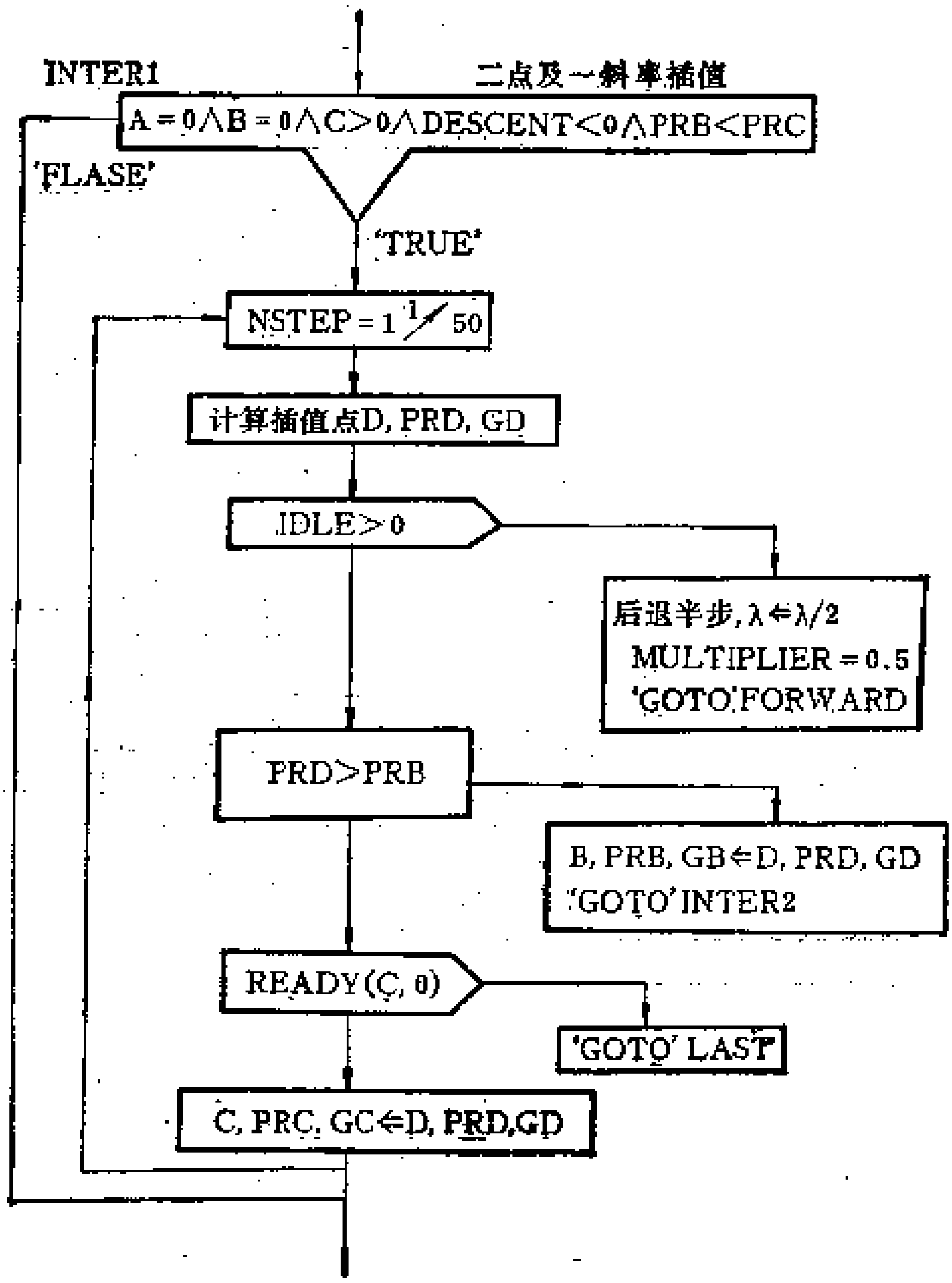
自点 POINT[1:N]出发，沿方向 S[1:N]线性搜索，找一个符合精度要求的最优参数B，使惩罚函数在点 POINT[1:N]+B*S[1:N]得极小化解。

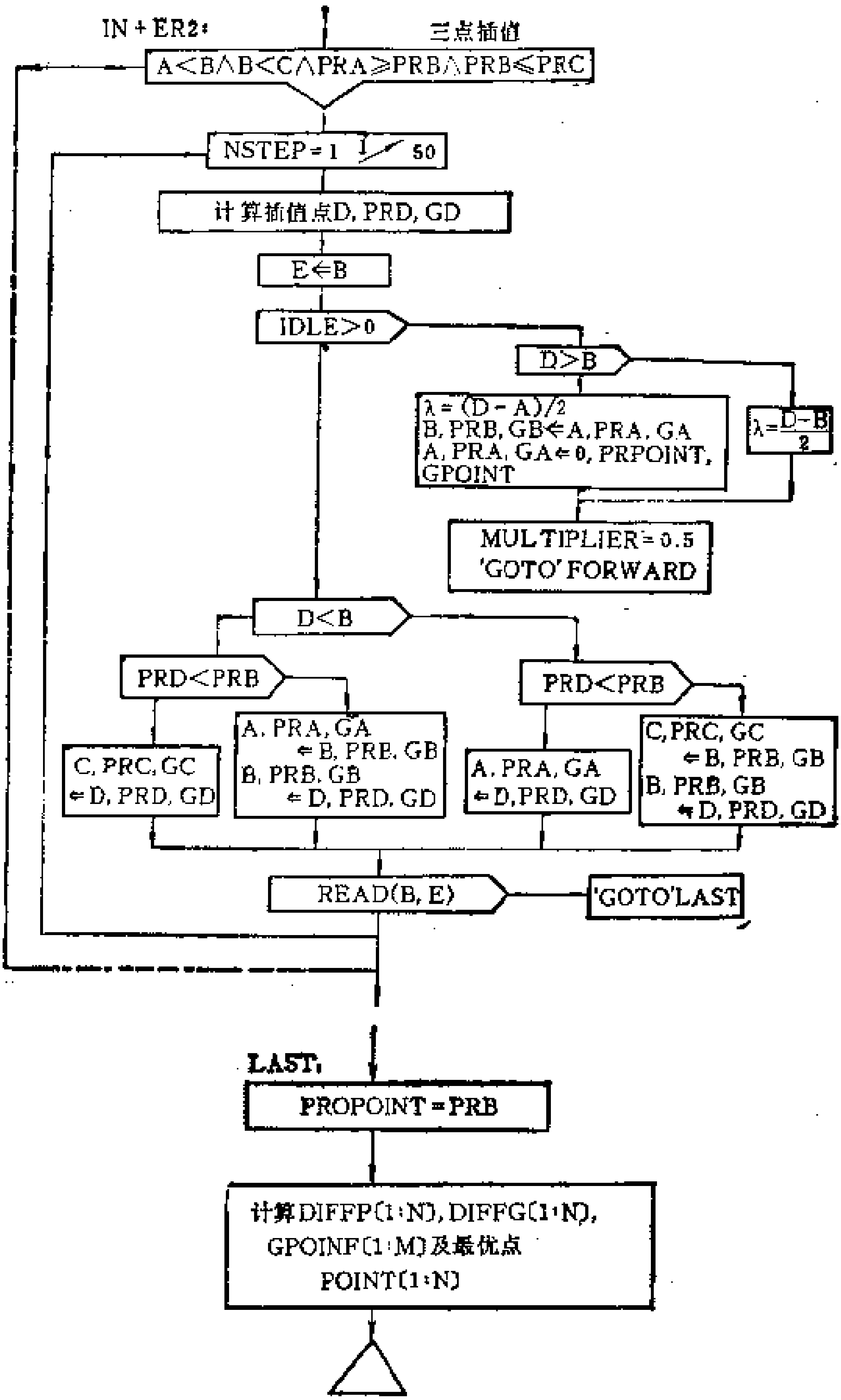
根据序列无约束极小化技巧，由 D．F．P．方法构造下降方向

S〔1:N〕。沿方向S〔1:N〕的极小化步骤是：开始沿方向S〔1:N〕移动，进行搜索；然后根据不同情况进行内插1（由二点及一斜率构成内插）及内插2（由三点构成内插）。内插点舍取的原则是，使留下来的点能包围极小化的点，这样反复进行内插，直到相继两点的每一分量之差的绝对值小于对应的相对精度与绝对精度，则最后一个迭代点就是线性极小化的近似点。

（c）示意框图：




INTER1
二点及一斜率插值
A = 0∧B = 0∧C>0∧DESCENT<0∧PRB<PRC
'FLASE'
'TRUE'
NSTEP = 1 1/50
计算插值点D, PRD, GD
IDLE>0
后退半步, λ⇐λ/2
MULTIPLIER = 0.5
'GOTO'FORWARD
PRD>PRB
B, PRB, GB⇐D, PRD, GD
'GOTO' INTER2
READY(C, 0)
'GOTO' LAST
C, PRC, GC⇐D, PRD, GD


IN+ER2:
三点插值
A<B∧B<C∧PRA≥PRB∧PRB≤PRC
NSTEP=1 1 50
计算插值点D, PRD, GD
E⇐B
IDLE>0
D>B
λ=(D-A)/2
B, PRB, GB⇐A, PRA, GA
A, PRA, GA⇐0, PRPOINT, GPOINT
λ=(D-B)/2
MULTIPLIER=0.5
'GOTO' FORWARD
D<B
PRD<PRB
PRD<PRB
C, PRC, GC ⇐D, PRD, GD
A, PRA, GA ⇐B, PRB, GB
B, PRB, GB ⇐D, PRD, GD
A, PRA, GA ⇐D, PRD, GD
C, PRC, GC ⇐B, PRB, GB
B, PRB, GB ⇐D, PRD, GD
READ(B, E)
'GOTO' LAST
LAST:
PROPOINT=PRB
计算DIFFP[1:N], DIFFG[1:N],
GPOINF[1:M]及最优点
POINT[1:N]

5）过程 RCYCLE

'PROCEDURE' RCYCLE (R, XR, UR, RAXR, AAXR)

（a）形式参数意义：

a）R 实型量

控制参数或惩罚因子。

b）XR 实型数组 XR[1:N]

存放循环的初始点及求得的极值点。

c）UR 实型数组 UR[1:M]

存放求解问题的对偶解。

d）RAXR 实型量

控制收敛的相对精度。

e）AAXR 实型量

控制收敛的绝对精度。

（b）功能：

对于固定值 r_k，求惩罚函数 $P_r(\mathbf{x})$ 的无约束极小化解 x_k^*。

从一开始，我们就介绍了本程序求解问题，分固定 r_k 与变动 r_k 两个步骤。惩罚函数 $P_r(\mathbf{x})$ 的极小化解按公式

$$\mathbf{x}^{k+1}=\mathbf{x}^k-b_k\mathbf{B}_k f(\mathbf{x}^k)$$

计算。过程 LINEMIN 讨论了如何选择下降因子 b_k，本过程就是要解决如何选择下降方向 $(-\mathbf{B}_k f(\mathbf{x}^k))$，从而解决了对于固定 r_k，如何求得惩罚函数 $P_r(\mathbf{x})$ 的无约束极小化解 x_k^* 的问题。

（c）示意框图（见下页）。

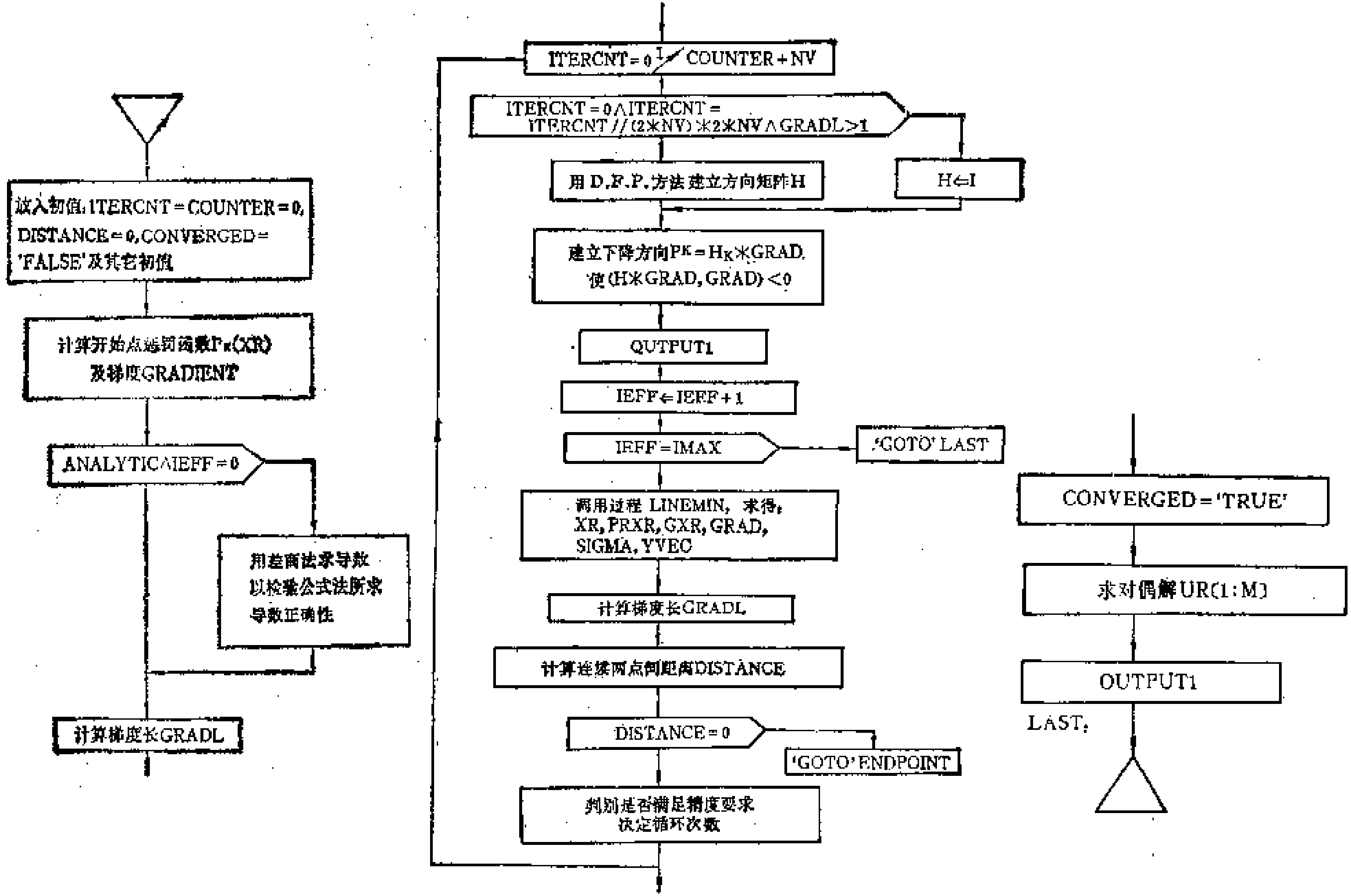
6）过程 EXTRAPOL

'PROCEDURE' EXTRAPOL(T, ORDER, DIM, NEW, RESULT, CONV)

（a）形式参数意义：

a）T 实型数组 T (1:DIM, 0:ORDER)

外推求解中用作工作单元。



放入初值:ITERCNT=COUNTER=0,
DISTANCE=0,CONVERGED=
'FALSE'及其它初值
计算开始点惩罚函数P_K(XR)
及梯度GRADIENT
ANALYTIC∧IEFF=0
用差商法求导数
以检验公式法所求
导数正确性
计算梯度长GRADL
ITERCNT=0
COUNTER+NV
ITERCNT=0∧ITERCNT=
ITERCNT//(2×NV)×2×NV∧GRADL>1
用D.F.P.方法建立方向矩阵H
H⇐I
建立下降方向PK=H_K×GRAD
使(H×GRAD,GRAD)<0
OUTPUT1
IEFF⇐IEFF+1
IEFF=IMAX
'GOTO' LAST
调用过程 LINEMIN，求得:
XR,PRXR,GXR,GRAD,
SIGMA,YVEC
计算梯度长GRADL
计算连续两点间距离DISTANCE
DISTANCE=0
'GOTO' ENDPOINT
判别是否满足精度要求
决定循环次数
CONVERGED='TRUE'
求对偶解UR(1:M)
OUTPUT1
LAST:

b）ORDER　整型量

外推的阶数，这里 ORDER≤6。

c）DIM　整型量

外推点的维数。

d）NEW　实型数组 NEW〔1:DIM〕

外推时输入的初始点值。

e）RESULT　实型数组 RESULT〔1:DIM〕

外推结果的最优点值。

f）CONV　布尔变量

控制满足收敛与否。

（b）功能：

对于每一个固定的惩罚因子 r_k，由过程 RCYCLE 求得的一个无约束极小化解 x_k^*，进行外推，以加速逼近于问题的最优解。

（c）示意框图（见下页）。

7）实函数过程 VECVEC 与 MATVEC

实函数过程 VECVEC（L，U，SHIFT，**A**，**B**）计算向量 **A** 与向量 **B** 的内积，即

$$\mathrm{VECVEC}=\sum_{K=L}^{U} \mathbf{A}_K \cdot \mathbf{B}_{SHIFT+K}$$

实函数过程 MATVEC（L，U，I，**A**，**B**）计算矩阵 **A** 的第 i 行向量与向量 **B** 的内积，即

$$\mathrm{MATVEC}=\sum_{K=L}^{U} \mathbf{A}_{IK}\mathbf{B}_K$$

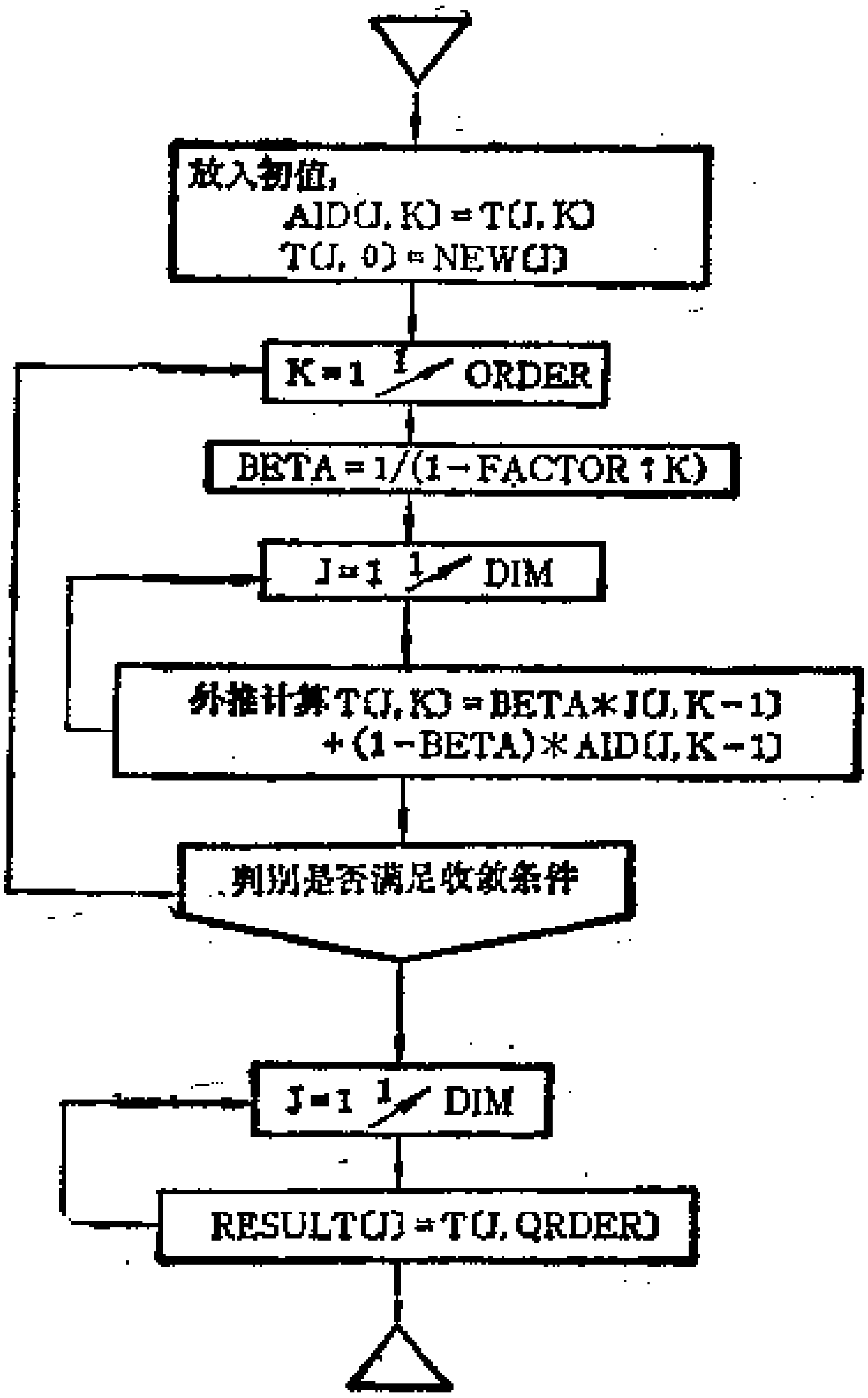
8）主过程 MINIMIZE

（a）形式参数意义：

前面已经介绍，从略。

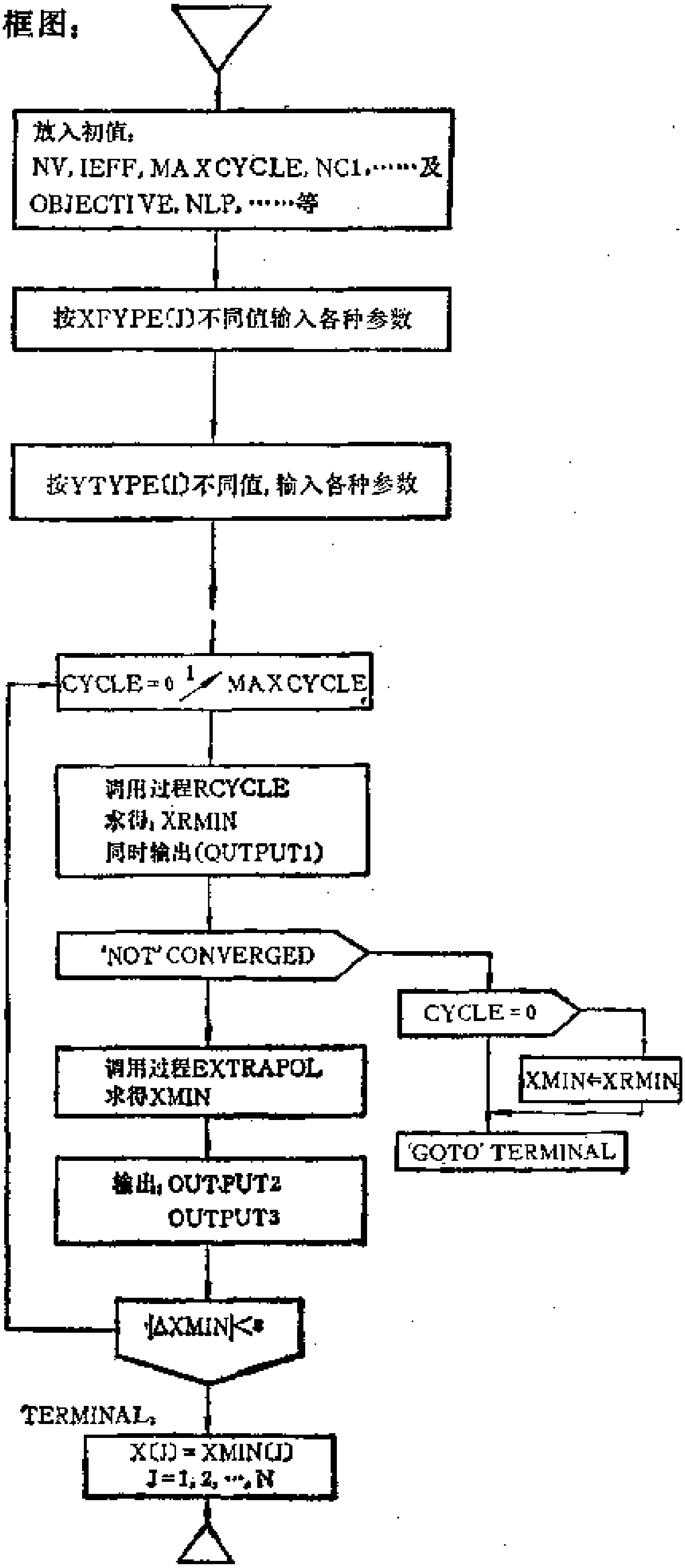
（b）功能：

前面已介绍，对于给定的目标函数及约束函数，可求解三类极小化问题。



放入初值,
AID[J, K] = T[J, K]
T[J, 0] = NEW[J]
K = 1 1 ORDER
BETA = 1/(1 - FACTOR ↑ K)
J = 1 1 DIM
外推计算 T[J, K] = BETA * J[J, K - 1]
+ (1 - BETA) * AID[J, K - 1]
判别是否满足收敛条件
J = 1 1 DIM
RESULT[J] = T[J, QRDER]

（c）示意框图：


放入初值：
NV, IEFF, MAXCYCLE, NC1,……及
OBJECTIVE, NLP,……等
按XFYPE(J)不同值输入各种参数
按YTYPE(J)不同值，输入各种参数
CYCLE=0 1 MAXCYCLE
调用过程RCYCLE
求得：XRMIN
同时输出(OUTPUT1)
'NOT' CONVERGED
CYCLE=0
XMIN←XRMIN
'GOTO' TERMINAL
调用过程EXTRAPOL
求得XMIN
输出：OUTPUT2
OUTPUT3
|ΔXMIN|<ε
TERMINAL:
X(J)=XMIN(J)
J=1, 2, …, N

9）输出过程 OUTPUT 1，OUTPUT 2，OUTPUT 3

（a）过程 OUTPUT1　各次迭代结果打印过程。

在调用过程 RCYCLE 时，即对于固定 r_k，求惩罚函数无约束极小化解 XRMIN 时，对于每次迭代数（ITERATION），求得的各种值进行打印，在程序中编写了下列三种输出。

短输出

短输出是对于每次迭代，输出单行的一览表，打印下列即时值：

a）ITERATION　迭代总次数。

b）PENALTY VALUE　惩罚函数即时值。

c）GRADIENT LENGTH　惩罚函数的梯度长度。

d）DISTANCE　即时迭代点与前面一迭代点之间的距离。

X——输出

每次迭代除打印短输出内容外，还输出：

a）VARIABLE　解向量X的各个分量的编号。

b）SOLUTION VECTOR　解向量的即时迭代点的值。

c）GRADIENT OF PEN. F.　惩罚函数的梯度的即时值。

d）GRAD〔DIFFERENCES〕　用差分法计算的惩罚函数初始点的梯度值。当 ANALYTIC 为真时，为检验所编导数过程的正确性，在初始点除打印求导过程所求导数值外，还打印用差分法所求的导数值，以进行比较。当 ANALYTIC 为假时，不打印。

G——输出

求解问题函数的输出：

a）FUNCTION　问题函数的编号。

b）FUNCTION VALUES　问题函数的即时值。

c）DUAL SOLUTION　对偶解。当有约束且收敛时，打印之。

根据主过程中参数COC的取值为1或2，以及开 始迭代收敛与否，分别打印短输出，X——输出或G——输出。

（b）过程OUTPUT 2　函数计算次数打印过程。

a）EVALUATIONS OF FUNCTIONS FOR NUM DIFF

利用差分法函数的计算次数，即NC_1值。

b）EVALUATIONS OF FUNCTIONS FOR LINE SEARCHES

线性搜索中函数计算次数，即NC_2值。

c）EVALUATIONS OF DERIVATIVES

外供求导数过程调用次数，即NC_3值。

d）AEQUIVALENT FUNCTION EVALUATIONS

计算函数相当总次数，即

$$NC_1+NC_2+NC_3\times NV$$

值。

e）EVALUATIONS OF FUNCTIONS PER LINF MINIMUM

每次一维线性极小化计算函数的平均次数，即NC_2/NSEARCH的值。

f）REJECTED POINTS BECAUSE OF CONSTRAINT VIOLATION

由于违反约束而报废点的个数，即NRP值。

（c）过程OUTPUT 3　外推后得解的打印过程。

a）EXTRAPOLATION, CYCLE*, ORDER*

CYCLE：表示无约束求极值循环的次数。

ORDER：外推的阶数。

b）VARIABLE SOLUTION VECTOR

对于固定R，外推后求得的最优解XMIN〔1:N〕。

c）FUNCTION VALUES

外推后，对应的最优点的问题函数的值。

d）DUAL SOLUTION

外推后，相应的对偶解的值。

4．如何调用主过程举例

要调用主过程

MINIMIZE (X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES, N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC, IMAX)

我们必须按求解问题的要求，对参数进行必要的说明，并输入各种初值。

（1）对形参必须用相应的实参来代替，并在程序一开始就进行说明。

（2）编写求解问题的函数过程 FUNCTIONS及求导数过程DERIVATIVES，若用差分法求导，过程 DERIVATIVES的过程体可用空语句表示。

（3）对于形式参数，按求解问题的要求，填入需要的数值。例如，希望用差分法求导数，则ANALYTIC取'FALSE'，只要短输出，则COC取1等。

（4）输入初始点的值，并按求解问题的要求输入XTYPE〔1:N〕，GTYPE〔1:M〕的值。

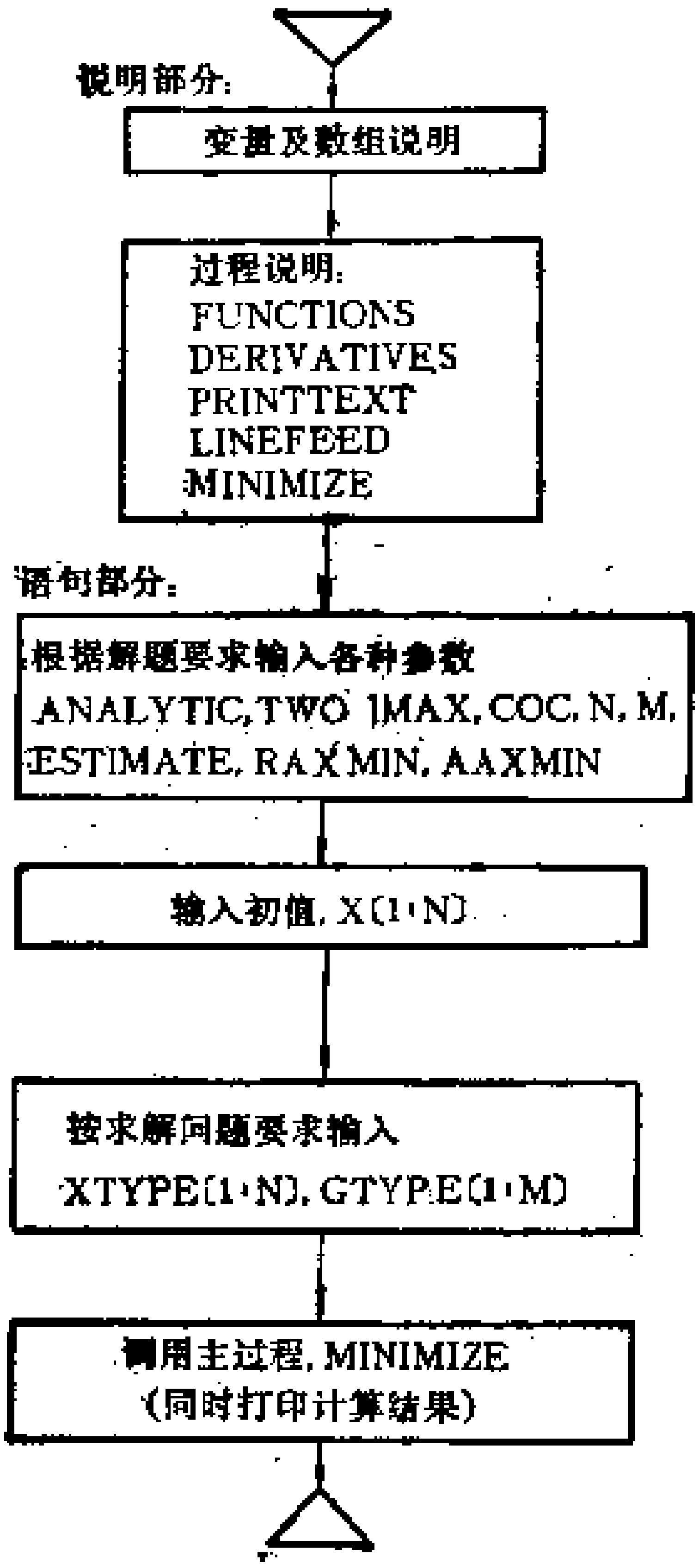
总程序的结构及解题过程，用下页框图示意。

总程序一开始引入两个整型量LINAGE与MPLUSN，它们的作用是：

1）LINAGE：控制打印行数用；

2）MPLUSN＝M＋N。

若总程序说明部分XTYPE〔1:N〕，GTYPE〔1:M〕用动态数组形式，则N.-M.必须在外层分程序中进行说明，并赋值。对于有些形式参数，如COC等，我们可以利用控制台变量，根据需要临时从控制台上输入，使程序使用更为方便。


说明部分:
变量及数组说明
过程说明:
FUNCTIONS
DERIVATIVES
PRINTTEXT
LINEFEED
MINIMIZE
语句部分:
根据解题要求输入各种参数
ANALYTIC, TWO, JMAX, COC, N, M,
ESTIMATE, RAXMIN, AAXMIN
输入初值, X[1:N]
按求解问题要求输入
XTYPE[1:N], GTYPE[1:M]
调用主过程, MINIMIZE
(同时打印计算结果)

例　求点集A(x_1, x_2, x_3)与点集B(x_4, x_5, x_6)之间的最短距离d(参见图7-2),

使合于

$$x_1^2+x_2^2+x_3^2\leqslant 5$$

$$(x_4-3)^2+x_5^2\leqslant 1$$

$$4\leqslant x_6\leqslant 8$$

初始点取(1,1,1,3,0,5)。

此例在本说明总程序结构中已介绍了。从几何上看，此题是求一个球体与一个圆柱体之间的最短距离。

求解问题可写成：

极小化　$d^2=g_5(\mathrm{x})=(x_1-x_4)^2+(x_2-x_5)^2+(x_3-x_6)^2$

使合于　$g_1(\mathrm{x})=5-x_1^2-x_2^2-x_3^2\geqslant 0$

$g_2(\mathrm{x})=1-(x_4-3)^2-x_5^2\geqslant 0$

$g_3(\mathrm{x})=8-x_6\geqslant 0$

$g_4(\mathrm{x})=x_6-4\geqslant 0$

由问题可知：

N=6，有六个变数。M=5，有五个函数。其中一个$g_5(\mathrm{x})$是目标函数，另外四个是约束函数。

1)按求解问题编写函数过程FUNCTIONS及导数过程DERIVATIVES。这两个过程，在前面总程序结构部分中已编了。

2)输入相应的参数值。

这里N=6，M=5，其它参数可根据不同要求，输入不同值。例如：

ANALYTIC取'FALSE'，TWO取'TRUE'，

IMAX=100，COC=1，ESTIMATE=5，

RAXMIN=AAXMIN=10^{-5}

3)输入XTYPE[1:N]，GTYPE[1:M]的各种值。

根据求解问题可知：

XTYPE[J]=1，(J=1,2,3,4,5,6)表示所有变

数X〔J〕都是自由变量。

GTYPE〔I〕= 2 （I = 1，2，3，4）表示 $g_1(x)$，$g_2(x)$，$g_3(x)$，$g_4(x)$ 都是不等式约束。

GTYPE〔5〕= 1 表示 $g_5(x)$ 是目标函数。

（4）输入初值。

X〔J〕= 1 （J = 1，2，3），X〔4〕= 3，X〔5〕= 0，X〔6〕= 5

计算程序见附录二。

计算结果得：

极小化点是：$x_1 = 1$，$x_2 = 0$，$x_3 = 2$，$x_4 = 2$，$x_5 = 0$，$x_6 = 4$。即球面上点P（1，0，2）与圆柱体底面上点Q(2，0，4)是两个几何体上距离最短的点，最短距离是$\sqrt{5}$。

上述结果，我们也可用拉格朗日乘数法计算来验证。

附录一 直接法计算程序

用FORTRAN语言书写的源程序：

```
      MASTER EXAMPLE
      INTEGER P
      DEMENSION X(50), X1(50,50), F(50),
     + X2(50,50), R(100), SUM(50), H(50),
     + SR(50), ROLD(100)
      COMMON DUM1, DUM2, DUM3, T, EPSILON
      PUBLIC NX, NC, NIC, STEP, IN, INF, K1,
     + K2, K3, K4, K5, K6, K7, K8, K9, FDIFER,
     + SEQL, X, X1, X2, R, SUM, F, SR, ROLD,
     + SCALE, FOLD, LFEAS, L5, L6, L7, L8,
     + L9, R1A, R2A, R3A
      READ(1) N, M, P, T, EPSILON
      IF(N.GT.50) GOTO 7777
      IF(P+1.GT.100) GOTO 7777
      NX=N
      NC=M
      NIC=P-M
10    READ(1) (X(I), I=1, N)
      WRITE(2, 11)
      CALL FLEXIPLEX
      INTEND=KEY(0, 1)
      IF(INTEND.EQ.0) GOTO 7777
      PAUSE 10
      GOTO 10
11    FORMAT(//////60X, 8H EXAMPLE/
     +                60X, 8H ********)
7777  STOP
      END
```

```
      SUBROUTINE PROBLEM(INQ)

      DIMENSION R(100),X(50)
      PUBLIC R,X
      GOTO (1,2,3),INQ
1     CONTINUE
      R(1)=25.-X(1)*X(1)-X(2)*X(2)
      GOTO 5
2     CONTINUE
      R(2)=10.*X(1)-X(1)*X(1)+10.*
     +      X(2)-X(2)*X(2)-34.
      R(3)=X(1)
      R(4)=X(2)
3     CONTINUE
      R(5)=4.*X(1)-X(2)*X(2)-12.
5     RETURN
      END

      SUBROUTINE SUMR

C   THIS SUBROUTINE COMPUTES THE SUM OF
C   THE SQUARE VALUES OF THE VIOLATED
C   CONSTRAINTS IN ORDER TO BE COMPARED
C   WITH THE TOLERANCE CRITERION
      DIMENSION R(100),X(50),SUM(50)
      PUBLIC NC,NIC,IN,SEQL,K7,K8,X,
     +      R,SUM
      SUB(IN)=0.
      CALL PROBLEM(2)
      SEQL=0.
      IF(NIC.EQ.0) GOTO 4
      DO 1 J=K7,K8
      IF(R(J).GE.0.) GOTO 1
      SEQL=SEQL+R(J)*R(J)
1     CONTINUE
4     IF(NC.EQ.0) GOTO 3
      CALL PROBLEM(1)
      DO 2 J=1,NC
2     SEQL=SEQL+R(J)*R(J)
3     SUM(IN)=SEQL
```

```
5       RETURN
        END

        SUBROUTINE START

        DIMENSION A(50,50),X(50),
       +          X1(50,50)
        PUBLIC NX,STEP,K1,X,X1
        VN=NX
        STEP1=STEP/(VN*SQRT(2.))*
       +      (SQRT(VN+1.)+VN-1.)
        STEP2=STEP/(VN*SQRT(2.))*
       +      (SQRT(VN+1.)-1.)
        DO 1 J=1,NX
1       A(1,J)=0
        DO 2 I=2,K1
        DO 4 J=1,NX
4       A(I,J)=STEP2
        L=I-1
        A(I,L)=STEP1
2       CONTINUE
        DO 3 I=1,K1
        DO 3 J=1,NX
3       X1(I,J)=X(J)+A(I,J)
        RETURN
        END

        SUBROUTINE FEASBL

C    SUBROUTINE FEASBL MINIMIZES THE SUM
C    OF THE SQUARE VALUES OF THE VILATED
C    CONSTRAINTS. IT IS CALLED EVERY TIME
C    THE COMBINED VALUE OF THE VIOLATED
C    CONSTRAINTS EXCEEDS THE VALUE OF THE
C    TOLERANCE CRITERION FOR THE CURRENT
C    STAGE
        DIMENSION X(50),X1(50,50),R(100),
       +   X2(50,50),SUM(50),F(50),H(50),
       +   SR(50),ROLD(100),R1(100),
       +   R2(100),R3(100),FLG(10)
```

```
      COMMON  DUM1, DUM2, DUM3, SIZE, DUM5
      PUBLIC  NX, NC, NIC, STEP, IN, INF,
     +   FDIFER, SEQL, K1, K2, K3, K4, K5,
     +   K6, K7, K8, K9, X, X1, X2, R, SUM, F,
     +   SR, ROLD, SCALE, FOLD, LFEAS, L5,
     +   L6, L7, L8, L9, R1A, R2A, R3A
      ALPHA=1.0
      BETA=0.5
      GAMMA=2.0
      XNX=NX
      ICONT=0
      LCHEK=0
      ICHEK=0
25    CALL  START
      DO  3  I=1, K1
      DO  4  J=1, NX
4     X(J)=X1(I, J)
      IN=I
      CALL  SUMR
3     CONTINUE
C   SELECT  LARGEST  VALUE  OF  SUM(I)  IN
C   SIMPLEX:
28    SUMH=SUM(1)
      INDEX=1
      DO  7  I=2, K1
      IF(SUM(I).LE.SUMH)  GOTO  7
      SUMH=SUM(I)
      INDEX=I
7     CONTINUE
C   SELECT  MINIMUM  VALUE  OF  SUM(I)  IN
    SIMPLEX:
      SUML=SUM(1)
      KOUNT=1
      DO  8  I=2, K1
      IF(SUML.LE.SUM(I))  GOTO  8
      SUML=SUM(I)
      KOUNT=I
8     CONTINUE
C   FIND  CENTROID  OF  POINTS  WITH  I
C   DIFFERENT  THAN  INDEX:
```

```
      DO 9 J=1,NX
      SUM2=0.
      DO 10 I=1,K1
10    SUM2=SUM2+X1(I,J)
      X1(K2,J)=1./XNX*(SUM2
     +            -X1(INDEX,J))
C   FIND REFLECTION OF HIGH POINT
    THROUGH CENTROID:
      X1(K3,J)=(1.+ALPHA)*X1(K2,J)
     +             -ALPHA*X1(INDEX,J)
9     X(J)=X1(K3,J)
      IN=K3
      CALL SUMR
      IF(SUM(K3).LT.SUML) GOTO 11
C   SELECT SECOND LARGEST VALUE IN
    SIMPLEX:
      IF(INDEX.EQ.1) GOTO 38
      SUMS=SUM(1)
      GOTO 39
38    SUMS=SUM(2)
39    DO 12 I=1,K1
      IF((INDEX-I).EQ.0) GOTO 12
      IF(SUM(I).LE.SUMS) GOTO 12
      SUMS=SUM(I)
12    CONTINUE
      IF(SUM(K3).GT.SUMS) GOTO 13
      DO 32 J=1,NX
32    X1(INDEX,J)=X1(K3,J)
      SUM(INDEX)=SUM(K3)
      GOTO 37
C   FORM EXPANSION OF NEW MINIMUM IF
C   REFLECTION HAS PRODUCED ONE
C   MINIMUM:
11    DO 15 J=1,NX
      X1(K4,J)=(1.-GAMMA)*X1(K2,J)
     +             +GAMMA*X1(K3,J)
15    X(J)=X1(K4,J)
      IN=K4
      CALL SUMR
      IF(SUM(K4).LT.SUML) GOTO 16
```

```
      GOTO 14
13    IF(SUM(K3).GT.SUMH) GOTO 17
      DO 18 J=1,NX
18    X1(INDEX,J)=X1(K3,J)
17    DO 19 J=1,NX
      X1(K4,J)=BETA*X1(INDEX,J)
     +         +(1.-BETA)*X1(K2,J)
19    X(J)=X1(K4,J)
      IN=K4
      CALL SUMR
      IF(SUMH.GT.SUM(K4)) GOTO 6
C  REDUCE SIMPLEX BY HALF IF REFLECTION
C  HAPPENS TO PRODUCE A LARGER VALUE
C  THAN THE MAXIMUM,
      DO 20 J=1,NX
      DO 20 I=1,K1
20    X1(I,J)=0.5*(X1(I,J)
     +         +X1(KOUNT,J))
      DO 29 I=1,K1
      DO 30 J=1,NX
30    X(J)=X1(I,J)
      IN=I
      CALL SUMR
29    CONTINUE
5     SUML=SUM(1)
      KOUNT=1
      DO 23 I=2,K1
      IF(SUML.LE.SUM(I)) GOTO 23
      SUML=SUM(I)
      KOUNT=I
23    CONTINUE
37    SR(INF)=SQRT(SUM(KOUNT))
      DO 27 J=1,NX
27    X(J)=X1(KOUNT,J)
      GOTO 26
6     DO 31 J=1,NX
31    X1(INDEX,J)=X1(K4,J)
      SUM(INDEX)=SUM(K4)
      GOTO 5
16    DO 21 J=1,NX
```

```
      X1(INDEX,J)=X1(K4,J)
21    X(J)=X1(INDEX,J)
      SUM(INDEX)=SUM(K4)
      SR(INF)=SQRT(SUM(K4))
      GOTO 26
14    DO 22 J=1,NX
      X1(INDEX,J)=X1(K3,J)
22    X(J)=X1(INDEX,J)
      SUM(INDEX)=SUM(K3)
      SR(INF)=SQRT(SUM(K3))
26    ICONT=ICONT+1
      DO 36 J=1,NX
36    X2(INF,J)=X(J)
      IF(ICONT.LT.2*K1) GOTO 50
      ICONT=0
      DO 24 J=1,NX
24    X(J)=X1(K2,J)
      IN=K2
      CALL SUMR
      DIFER=0.
      DO 57 I=1,K1
57    DIFER=DIFER+(SUM(I)-SUM(K2))**2
      DIFER=1/(K7*XNX)*SQRT(DIFER)
      IF(DIFER.GT.1.0E-14) GOTO 50
C  IF FLEXIBLE SIMPLEX METHOD FAILED
C  TO SATISFY THE CONSTRAINTS WITHIN
C  THE TOLERANCE CRITERION FOR THE
C  CURRENT STAGE, THE SEARCH IS PER
C  TURBED FROM THE POSITION WHERE THE
C  X VECTOR IS STUCK AND THEN FEASBL
C  IS REPEATED ONCE MORE THE BEGINNING
51    IN=K1
      STEP=20.*FDIFER
      CALL SUMR
      SR(INF)=SQRT(SEQL)
      DO 52 J=1,NX
52    X1(K1,J)=X(J)
      DO 53 J=1,NX
      FACTOR=1.
      X(J)=X1(K1,J)+FACTOR*STEP
```

```
      X1(L9,J)=X(J)
      IN=L9
      CALL SUMR
      X(J)=X1(K1,J)-FACTOP*STEP
      X1(L5,J)=X(J)
      IN=L5
      CALL SUMR
56    IF(SUM(L9).LT.SUM(K1)) GOTO 54
      IF(SUM(L5).LT.SUM(K1)) GOTO 55
      GOTO 97
54    X1(L5,J)=X1(K1,J)
      SUM(L5)=SUM(K1)
      X1(K1,J)=X1(L9,J)
      SUM(K1)=SUM(L9)
      FACTOR=FACTOR+1.
      X(J)=X1(K1,J)+FACTOR*STEP
      IN=L9
      CALL SUMR
      GOTO 56
55    X1(L9,J)=X1(K1,J)
      SUM(L9)=SUM(K1)
      X1(K1,J)=X1(L5,J)
      SUM(K1)=SUM(L5)
      FACTOR=FACTOR+1.
      X(J)=X1(K1,J)-FACTOR*STEP
      IN=L5
      CALL SUMR
      GOTO 56
C   ONE DIMENSIONAL SEARCH BY GOLDEN
C   SECTION ALONG EACH COORDINATE
97    H(J)=X1(L9,J)-X1(L5,J)
      X1(L6,J)=X1(L5,J)+H(J)*R1A
      X(J)=X1(L6,J)
      IN=L6
      CALL SUMR
      X1(L7,J)=X1(L5,J)+H(J)*R2A
      X(J)=X1(L7,J)
      IN=L7
      CALL SUMR
      IF(SUM(L6).GT.SUM(L7)) GOTO 68
```

```
      X1(L8,J)=X1(L5,J)+(1.-R3A)*H(J)
      X1(L5,J)=X1(L7,J)
      X(J)=X1(L8,J)
      IN=L8
      CALL SUMR
      IF(SUM(L8).GT.SUM(L6)) GOTO 76
      X1(L5,J)=X1(L6,J)
      SUM(L5)=SUM(L6)
      GOTO 75
76    X1(L9,J)=X1(L8,J)
      SUM(L9)=SUM(L8)
      GOTO 75
68    X1(L9,J)=X1(L6,J)
      X1(L8,J)=X1(L5,J)+R3A*H(J)
      X(J)=X1(L8,J)
      IN=L8
      CALL SUMR
      STEP=SIZE
      SUM(L9)=SUM(L6)
      IF(SUM(L7).GT.SUM(L8)) GOTO 71
      X1(L5,J)=X1(L8,J)
      SUM(L5)=SUM(L8)
      GOTO 75
71    X1(L9,J)=X1(L7,J)
      SUM(L9)=SUM(L7)
75    IF(ABS(X1(L9,J)-X1(L5,J)).GT.
         0.01*FDIFER) GOTO 97
      X1(K1,J)=X1(L7,J)
      X(J)=X1(L7,J)
      SUM(K1)=SUM(L5)
      SR(INF)=SQRT(SUM(K1))
      IF(SR(INF).LT.FDIFER) GOTO 760
53    CONTINUE
      ICHEK=ICHEK+1
      STEP=FDIFER
      IF(ICHEK.LE.2) GOTO 25
      FOLD=1.0E-12
      WRITE(2,853)
      WRITE(2,850)
      WRITE(2,851) (X(J),J=1,NX)
```

```
      WRITE(2,852) FDIFER,SR(INF)
      GOTO 46
760   DO 761 J=1,NX
      X2(INF,J)=X1(K1,J)
761   X(J)=X1(K1,J)
50    IF(SR(INF).GT.FDIFER) GOTO 28
C   MODIFIED LAGRANGE INTERPOLATION
C   FOR TIGHT INEQUALITIES
      IF(SR(INF).GT.0.) GOTO 35
      CALL PROBLEM(3)
      FINT=R(K9)
      DO 139 J=1,NX
139   X(J)=X2(INF,J)
      CALL PROBLEM(2)
      DO 40 J=K7,K8
40    R1(J)=R(J)
      DO 41 J=1,NX
41    X(J)=X1(KOUNT,J)
      CALL PROBLEM(2)
      DO 42 J=K7,K8
42    R3(J)=R(J)
      DO 43 J=1,NX
      H(J)=X1(KOUNT,J)-X2(INF,J)
43    X(J)=X2(INF,J)+0.5*H(J)
      CALL PROBLEM(2)
      FLG(1)=0.
      FLG(2)=0.
      FLG(3)=0.
      DO 44 J=K7,K8
      IF(R3(J).GE.0.) GOTO 44
      FLG(1)=FLG(1)+R1(J)*R1(J)
      FLG(2)=FLG(2)+R(J)*R(J)
      FLG(3)=FLG(3)+R3(J)*R3(J)
44    CONTINUE
      SR(INF)=SQRT(FLG(1))
      IF(SR(INF).LT.FDIFER) GOTO 35
      ALPHA1=FLG(1)-2.*FLG(2)+FLG(3)
      BETA1=3.*FLG(1)-4.*FLG(2)+FLG(3)
      RATIO=BETA1/(4.*ALPHA1)
      DO 45 J=1,NX
```

```
45      X(J)=X2(INF,J)+H(J)*RATIO
        IN=INF
        CALL SUMR
        SR(INF)=SQRT(SEQL)
        IF(SR(INF).LT.FDIFER) GOTO 465
        DO 49 I=1,20
        DO 48 J=1,NX
48      X(J)=X(J)-0.05*H(J)
        CALL SUMR
        SR(INF)=SQRT(SEQL)
        IF(SR(INF).LT.FDIFER) GOTO 465
49      CONTINUE
465     CALL PROBLEM(3)
        IF(FINT.GT.R(K9)) GOTO 46
        SR(INF)=0.
        GOTO 35
46      DO 47 J=1,NX
47      X2(INF,J)=X(J)
35      CONTINUE
        DO 335 J=1,NX
335     X(J)=X2(INF,J)
850     FORMAT(//16X,15H IT IS NOT POSS,
       +29HIBLE TO SATISFY THE VIOLATED,
       +28HCONSTRAINT SET FROM THIS VEC,
       +28HTOR. THE SEARCH WILL BE TERM,
       +8HINATED. /17X,12HPLEASE CHOOSE,
       +28H A NEW STARTING VECTOR AND R,
       +22HEPEAT SOLUTION AGAIN .)
851     FORMAT(//16X,15H THE VECTOR FOR,
       +28H WITCH THE CONSTRAINTS COULD,
       +20H NOT BE SATISFIED IS/(E32.6,
       +7E16.6))
852     FORMAT(//16X,15H THE TOLERANCE,
       +12HCRITERION = ,E14.6/16X,
       +28H THE SQUARE ROOT OF THE CONS,
       +21HTRAINTS SQUARED IS = ,E16.6)
853     FORMAT(//40X,15H * * * * * * * ,
       +28HSUBROUTINE FEASBL FAILS TO F,
       +28HIND A FEASIBLE POINT * * * *,
       +6H* * *)
```

```
        RETURN
        END

        SUBROUTINE WRITEX

        DIMENSION R(100), X(50)
        PUBLIC NX, NC, NIC, K6, K7, K9, X, R
        CALL PROBLEM(3)
        WRITE(2, 1) R(K9)
1       FORMAT(/16X, 16H OBJECTIVE FUNCT,
       +12HION VALUE = , E17.7)
        WRITE(2, 2) (X(I), I=1, NX)
2       FORMAT(/16X, 16H THE INDEPENDENT,
       +13H VECTORS ARE /(E46.7, 5E17.7))
        IF(NC.EQ.0) GOTO 6
        CALL PROBLEM(1)
        WRITE(2, 3) (R(J), J=1, NC)
3       FORMAT(/16X, 16H THE EQUALITY CO,
       +20HNSTRAINT VALUES ARE /
       + (E46.7, 5E17.7))
6       IF(NIC.EQ.0) GOTO 5
        CALL PROBLEM(2)
        WRITE(2, 4) (R(J), J=K7, K6)
4       FORMAT(/16X, 16H THE INEQUALITY ,
       +21HCONSTRAINT VALUE ARE /
       + (E46.7, 5E17.7))
5       RETURN
        END

        SUBROUTINE FLEXIPLEX

C   FLEXIPLEX SOLVES THE GENERAL NONLI-
C   NEAR PROGRAMMING PROBLEM.
C   NOMENCLATURE;
C   NX        TOTAL NUMBER OF INDEPENDENT
C             VARIABLES
C   NC        TOTAL NUMBER OF EQUALITY
C             CONSTRAINTS
C   NIC       TOTAL NUMBER OF INEQUALITY
C             CONSTRAINTS
```

```
C   SIZE     EDGE LENGTH OF THE INITIAL
C      .     POLYHEDRON
C   CONVER   CONVERGENCE CRITERION FOR
C            TERMINATION OF THE SEARCH
C   ALPHA    THE REFLECTION COEFFICIENT
C   BETA     THE CONTRACTION COEFFICIENT
C   GAMMA    THE EXPANSION COEFFICIENT
C   X(I)     THE ASSUMED VECTOR TO INI-
C            TIATE THE SEARCH
C   FDIFER   THE TOLERANCE CRITERION FOR
C            CONSTRAINT VIOLATION
C   ICONT    A COUNTER TO RECORD STAGE
C            COMPUTATIONS
C   NCONT    A COUNTER TO PRINT INFORMA-
C            TION EVERY (NX+1) STAGE
C   LOW      AN INDEX TO IDENTIFY INFOR-
C            MATION RELATED TO THE LOWEST
C            VALUE OF OBJ. FUNCTION IN
C            MOCT RECENT POLYHEDRON
C   LIGH     ANINDEX TO IDENTIFY INFOR-
C            MATION RELATED TO LARGEST
C            VALUE OF OBJ. FUNCTION IN
C            MOST RECENT POLYHEDRON
C   LSEC     AN INDEX TO IDENTIFY INFOR-
C            MATION RELATED TO THE SECOND
C            LARGEST VALUE OF OBJ. FUNC.
C            IN MOST RECENT POLYHEDRON
C   SUPPORTING SUBPROGRAMS, PROBLEM,
C      SUMR, START, FEASBL, AND WRITEX
       DIMENSION X(50), X1(50,50), F(50),
     +   X2(50,50), R(100), SUM(50),
     +   SR(50), ROLD(100), H(50)
       COMMON ALPHA, BETA, GAMMA, SIZE,
     +   CONVER
       PUBLIC NX, NC, NIC, STEP, IN, INF, F,
     +   FDIFER, SEQL, K1, K2, K3, K4, K5, K6,
     +   K7, K8, K9, X, X1, X2, R, SUM, SR, L9,
     +   ROLD, SCALE, FOLD, LFEAS, L5, L6,
     +   L7, L8, R1A, R2A, R3A
C   TO SELECT THE VALUES FOR COEFFICI-
```

```
C     ENTS ALPHA, BETA, AND GAMMA,
          ALPHA=1.0
          BETA=0.5
          GAMMA=2.0
          SIEP=SIZE
          WRITE(2,106)
          WRITE(2,759)
          WRITE(2,756)  NX,NC,NIC,SIZE,
         +              CONVER
          K1=NX+1
          K2=NX+2
          K3=NX+3
          K4=NX+4
          K5=NX+5
          K6=NC+NIC
          K7=NC+1
          K8=NC+NIC
          K9=K8+1
          N=NX-NC
          N1=N+1
          IF(N1.GE.3)  GOTO 50
          N1=3
          N=2
50        N2=N+2
          N3=N+3
          N4=N+4
          N5=N+5
          N6=N+6
          N7=N+7
          N8=N+8
          XN=N
          XNX=NX
          XNI=N1
          R1A=0.5*(SQRT(5.)-1.)
          R2A=R1A*R1A
          R3A=R2A*R1A
          L5=NX+5
          L6=NX+6
          L7=NX+7
          L8=NX+8
```

```
      L9=NX+9
      ICONT=1
      NCONT=1
      WRITE(2,115)
      WRITE(2,116) (X(J),J=1,NX)
      FDIFER=2.*(NC+1)*STEP
      FOLD=FDIFER
      IN=N1
      CALL SUMR
      SR(N1)=SQRT(SEQL)
      WRITE(2,763) FDIFER,SR(N1)
      IF(SR(N1).LT.FDIFER) GOTO 341
      CALL WRITEX
      WRITE(2,757)
      INF=N1
      STEP=0.05*FDIFER
      CALL FEASBL
      WRITE(2,764)
      WRITE(2,116) (X2(INF,J),J=1,NX)
      WRITE(2,765) SR(INF)
      IF(FOLD.LT.1.0E-09) GOTO 80
341   WRITE(2,35)
      WRITE(2,758) ICONT,FDIFER
      CALL WRITEX
      FTER=R(K9)
C   COMPUTE CENTROID OF ALL VERTICES
C   OF INITIAL POLYHEDRON
237   STEP1=STEP*(SQRT(XNX+1.)+XNX-1.)
     +        /(XNX*SQRT(2.))
      STEP2=STEP*(SQRT(XNX+1.)/(XNX*
     +        SQRT(2.))
      ETA=(STEP1+(XNX-1.)*STEP2)
     +      /XNX+1.)
      DO 4 J=1,NX
      X(J)=X(J)-ETA
4     CONTINUE
      CALL START
      DO 9 I=1,N1
      DO 9 I=1,NX
      X2(I,J)=X1(I,J)
```

```
9       CONTINUE
        DO 5 I=1,N1
        IN=I
        DO 6 J=1,NX
6       X(J)=X2(I,J)
        CALL SUMR
        SR(I)=SQRT(SEQL)
        IF(SR(I).LT.FDIFER) GOTO 8
        CALL FEASBL
        IF(FOLD.LT.1.0E-09) GOTO 80
8       CALL PROBLEM(3)
        F(I)=R(K9)
5       CONTINUE
1000    STEP=0.05*FDIFER
        ICONT=ICONT+1
C   SELECT LARGEST VALUE OF OBJ. FUNC.
C   FROM POLYHEDRON VERTICES
        FH=F(1)
        LHIGH=1
        DO 16 I=2,N1
        IF(F(I).LT.FH) GOTO 16
        FH=F(I)
        LHIGH=I
16      CONTINUE
C   SELECT MINIMUM VALUE OF OBJ. FUNC.
C   FROM POLYHEDROM VERTICES
41      FL=F(1)
        LOW=1
        DO 17 I=2,N1
        IF(FL.LT.F(I)) GOTO 17
        FL=F(I)
        LOW=I
17      CONTINUE
        DO 86 J=1,NX
86      X(J)=X2(LOW,J)
        IN=LOW
        CALL SUMR
        SR(LOW)=SQRT(SEQL)
        IF(SR(LOW).LT.1.0E-09) GOTO87
        INF=LOW
```

```
      CALL FEASBL
      IF(FOLD.LT.1.0E-09) GOTO 80
      CALL PROBLEM(3)
      F(LOW)=R(K9)
      GOTO 41
87    CONTINUE
C  FIND CENTROID OF POINTS WITH I
C  DIFFERENT THAN LHIGH
      DO 19 J=1,NX
      SUM2=0.
      DO 20 I=1,N1
20    SUM2=SUM2+X2(I,J)
19    X2(N2,J)=1./XN*(SUM2-
     +          X2(LHIGH,J))
      SUM2=0.
      DO 36 I=1,N1
      DO 36 J=1,NX
      SUM2=SUM2+(X2(I,J)-X2(N2,J))**2
36    CONTINUE
      FDIFER=(NC+1)/XN1*SQRT(SUM2)
      IF(FDIFER.LT.FOLD) GOTO 98
      FDIFER=FOLD
      GOTO 198
98    FOLD=FDIFER
198   CONTINUE
      FTER=F(LOW)
137   NCONT=NCONT+1
      IF(NCONT .LT.4*N1) GOTO37
      IF(ICONT.LT.1500) GOTO 337
      FOLD=O.5*FOLD
337   NCONT=0
      IPRINT=KEY(47,47)
      IF(IPRINT.EQ.1) GOTO 37
      WRITE(2,35)
      WRITE(2,735) ICONT,FDIFER
      CALL WRITEX
      IF(FDIFER.LT.CONVER) GOTO 81
C SELECT SECOND LARGEST VALUE OF OBJ.
C FUNCTION
      IF(LHIGH.EQ.1) GOTO 43
```

```
      FS=F(1)
      LSEC=1
      GOTO 44
43    FS=F(2)
      LSEC=2
44    DO 18 I=1,N1
      IF(LHIGH.EQ.I) GOTO 18
      IF(F(I).LT.FS) GOTO 18
      FS=F(I)
      LSEC=I
18    CONTINUE
C   REFLECT HIGH POINT THROUGH CENTROID
      DO 61 J=1,NX
      X2(N3,J)=(1.+ALPHA)*X2(N2,J)
     +           -ALPHA*X2(LHIGH,J)
61    X(J)=X2(N3,J)
      IN=N3
      CALL SUMR
      SR(N3)=SQRT(SEQL)
89    IF(SR(N3).LT.FDIFER) GOTO 82
      INF=N3
      CALL FEASBL
      IF(FOLD.LT.1.0E-09) GOTO 80
82    CALL PROBLEM(3)
      F(N3)=R(K9)
      IF(F(N3).LT.F(LOW)) GOTO 84
      IF(F(N3).LT.F(LSEC)) GOTO 92
      GOTO 60
92    DO 93 J=1,NX
93    X2(LHIGH,J)=X2(N3,J)
      SR(LHIGH)=SR(N3)
      F(LHIGH)=F(N3)
      GOTO 1000
C   EXPAND VECTER OF SEARCH ALONG
C   DIRECTION THROUGH CENTROID AND
C   REFLECTED VECTOR
84    DO 23 J=1,NX
      X2(N4,J)=(1.-GAMMA)*X2(N2,J)
     +           +GAMMA*X2(N3,J)
23    X(J)=X2(N4,J)
```

```
      IN=N4
      CALL SUMR
      SR(N4)=SQRT(SEQL)
      IF(SR(N4).LT.FDIFER) GOTO 25
      INF=N4
      CALL FEASBL
      IF(FOLD.LT.1.0E-09) GOTO 80
25    CALL PROBLEM(3)
      F(N4)=R(K9)
      IF(F(LOW).LT.F(N4)) GOTO 92
      DO 26 J=1,NX
26    X2(LHIGH,J)=X2(N4,J)
      F(LHIGH)=F(N4)
      SR(LHIGH)=SR(N4)
      GOTO 1000
60    IF(F(N3).GT.F(LHIGH)) GOTO 64
      DO 65 J=1,NX
65    X2(LHIGH,J)=X2(N3,J)
64    DO 66 J=1,NX
      X2(N4,J)=BETA*X2(LHIGH,J)
     +            +(1.-BETA)*X2(N2,J)
66    X(J)=X2(N4,J)
      IN=N4
      CALL SUMR
      SR(N4)=SQRT(SEQL)
      IF(SR(N4).LT.FDIFER) GOTO 67
      INF=N4
      CALL FEASBL
      IF(FOLD.LT.1.0E-09) GOTO 80
67    CALL PROBLEM(3)
      F(N4)=R(K9)
      IF(F(LHIGH).GT.F(N4)) GOTO 68
      DO 69 J=1,NX
      DO 69 I=1,N1
69    X2(I,J)=0.5*(X2(I,J)+X2(LOW,J))
      DO 70 I=1,N1
      DO 71 J=1,NX
      X(J)=X2(I,J)
      IN=I
      CALL SUMR
```

```
      SR(I)=SQRT(SEQL)
      IF(SR(I).LT.FDIFER) GOTO 72
      INF=I
      CALL FEASBL
      IF(FOLD.LT.1.0E-09) GOTO 80
72    CALL PROBLEM(3)
70    F(I)=R(K9)
      GOTO 1000
68    DO 73 J=1,NX
73    X2(LHIGH,J)=X2(N4,J)
      SR(LHIGH)=SR(N4)
      F(LHIGH)=F(N4)
      GOTO 1000
81    WRITE(2,760) ICONT,FDIFER
      CALL WRITEX
      WRITE(2,761)
      RETURN
80    WRITE(2,760) ICONT,FDIFER
      CALL WRITEX
      WRITE(2,762)
      RETURN
35    FORMAT(/57X,46H------------------,
     +----------------------------------)
106   FORMAT(////65X,13HTHE FLEXIBLE ,
     +16HTOLERANCE METHOD)
115   FORMAT(//16X,15HTHE STARTING VE,
     +28HCTOR SELECTED BY DESIGNER IS)
116   FORMAT(E46.7,5E17.7)
756   FORMAT(//26X,15H NUMBER OF INDE,
     +25HPENDENT VARIABLES        ,I5
     +/26X,23H NUMBER OF EQUALITY CON,
     +17HSTRAINTS         ,I5/26X,
     +28H NUMBER OF INEQUALITY CONSTR,
     +12HAINTS      ,I5/26X,6H SIZE ,
     +28HOF INITIAL POLYHEDRON       ,
     +6H      ,E12.5/26X, 9H THE DESI,
     +28HRED CONVERGENCE IS          ,
     +3H   ,E12.5/)
757   FORMAT(//16X,15H THE INITIAL X ,
     +28HVECTOR DOES NOT SATISFY THE ,
```

```
     ↑28HINITIAL TOLERANCE CRITERION.)
758   FORMAT(//26X,15H STAGE CALCULAT,
     ↑12HION NUMBER =,I5/26X,5H THE ,
     ↑22HTOLERANCE CRITERION = ,E14.6)
759   FORMAT(65X,14H* * * * * * * ///)
760   FORMAT(////16X,13H TOTAL NUMBER,
     ↑26H OF STAGES CALCULATIONS = ,
     ↑I5/16X,21H THE CONVERGENCE LIMI,
     ↑4HT = ,E14.6)
761   FORMAT(//68X,15H THESE ARE FINA,
     ↑11HL ANSWERS •)
762   FORMAT(//68X,15H THESE ARE NOT ,
     ↑15HFINAL ANSWERS •)
763   FORMAT(//26X,15H THE INITIAL TO,
     ↑28HLERANCE CRITERION IS        ,
     ↑E12.5/26X,18H THE SUM OF VIOLAT,
     ↑22HED CONSTRAINTS IS      ,E12.5)
764   FORMAT(//16X,15H THE VECTOR FOU,
     ↑28HND BY PROGRAM WHICH SATISFIE,
     ↑28HS THE INITIAL TOLERANCE IS )
765   FORMAT(/16X,16H SUM OF VIOLATED,
     ↑15H CONSTRAINTS = ,E17.7)
      RETURN
      END
      FINISH
```

附录二　解析法计算程序

用 ALGOL 语言书写的源程序

```
·BEGIN·
·INTEGER·N, M, COC, IMAX, I, J;
·INTEGER·MPLUSN, LINAGE;
·REAL·RAXMIN, AAXMIN, ESTIMATE;
·BOOLEAN·ANALYTIC, TWO, CONVERGED;
·INTEGER··ARRAY·XTYPE[1:6], GTYPE[1:5];
·REAL··ARRAY·X[1:6], GX[1:5], DGDX[1:5, 1:6];

·PROCEDURE·FUNCTIONS(X, GX);
·REAL··ARRAY·X, GX;
·BEGIN·GX[1]:=5-X[1]↑2-X[2]↑2-X[3]↑2;
 GX[2]:=1-(X[4]-3)↑2-X[5]↑2;
 GX[3]:=8-X[6];
 GX[4]:=X[6]-4;
 GX[5]:=(X[1]-X[4])↑2+(X[2]-X[5])↑2+(X[3]-X[6])↑2
·END·;

·PROCEDURE·DERIVATIVES(X, DGDX);
·REAL··ARRAY·X, DGDX;
·BEGIN·
·FOR·J:=1, 2, 3·DO·DGDX[1, J]:=-2*X[J];
 DGDX[2, 4]:=-2*(X[4]-3);
 DGDX[2, 5]:=-2*X[5];
 DGDX[3, 6]:=-1; DGDX[4, 6]:=1;
·FOR·J:=1, 2, 3·DO·DGDX[5, J]:=2*(X[J]-X[J+3]);
·FOR·J:=1, 2, 3·DO·DGDX[5, J+3]:=-2*(X[J]-X[J+3])
·END·;

·PROCEDURE·PRINTTEXT(N1, ST1, N2, ST2, N3, ST3, N4, ST4, N5);
·VALUE·N1, N2, N3, N4, N5;
```

```
•INTEGER•N1, N2, N3, N4, N5;
•STRING•ST1, ST2, ST3, ST4;
•BEGIN•
•INTEGER•I, N;
•SWITCH•SW:=L1, L2, L3, L4, ØVER;
 I:=0;
•FØR•N:=N1, N2, N3, N4, N5•DØ•
•BEGIN•
 I:=I+1;
•IF•N=0 •THEN• •ELSE•SPACE(N);
•GØTØ•SW[I];
 L1:PUTS(ST1); •GØTØ•ØVER;
 L2:PUTS(ST2); •GØTØ•ØVER;
 L3:PUTS(ST3); •GØTØ•ØVER;
 L4:PUTS(ST4);
 ØVER:•END•
•END•;

•PRØCEDURE•LINEFEED(N);
•VALUE•N;
•INTEGER•N;
•BEGIN•
•INTEGER•I;
•SWITCH•SWL:=PASS;
•FØR•I:=1•STEP•1•UNTIL•N•DØ•
•BEGIN•
 LINAGE:=LINAGE+1;
•IF•LINAGE=52•THEN• •ELSE• •GØTØ•PASS;
 PAGE; LINAGE:=3;
 LINE; LINE;
 PASS:LINE;
•END•
•END•;

•PRØCEDURE•MINIMIZE(X, FUNCTIØNS, XTYPE, GTYPE, ANALYTIC,
 DERIVATIVES, N, M, TWØ, RAXMIN, AAXMIN, ESTIMATE, CØNVE-
 RGED, CØC, IMAX);
•VALUE•N, M, RAXMIN, AAXMIN, ESTIMATE, CØC, IMAX;
•INTEGER•N, M, CØC, IMAX;
•REAL•RAXMIN, AAXMIN, ESTIMATE;
```

```
•BOOLEAN•ANALYTIC, TWO, CONVERGED,
•REAL• •ARRAY•X,
•INTEGER• •ARRAY•XTYPE, GTYPE,
•PROCEDURE•FUNCTIONS, DERIVATIVES;
•BEGIN•
•REAL•INITR, FACTOR;
•INTEGER•I, J, K, NV, IEFF, CYCLE, MAXCYCLE, NC1, NC2, NC3,
 NSEARCH, NRP,
•BOOLEAN•OBJECTIVE, CONSTRAINTS, NLP, HEADING, NONNEG;
•REAL• •ARRAY•XRMIN, XMIN[1:N], URMIN, UMIN[1:M],XTABLE[1:N,
 0:6], UTABLE[1:M, 0:6];
•BOOLEAN• •ARRAY•LOGP[1:MPLUSN],
•SWITCH•SWM:=TERMINAL;
•REAL• •PROCEDURE•VECVEC (L, U, SHIFT, A, B);
•VALUE•L, U, SHIFT;
•INTEGER•L, U, SHIFT;
•REAL• •ARRAY•A, B;
•BEGIN•
•INTEGER•K;
•REAL•S;
 S:=0.0;
•FOR•K:=L•STEP•1•UNTIL•U•DO•
 S:=A[K]*B[SHIFT+K]+S;
 VECVEC:=S
•END•;

•REAL• •PROCEDURE•MATVEC (L, U, I, A, B);
•VALUE•L, U, I;
•INTEGER•L, U, I;
•REAL• •ARRAY•A, B;
•BEGIN•
•INTEGER•K;
•REAL•S;
 S:=0.0;
•FOR•K:=L•STEP•1•UNTIL•U•DO•
 S:=A[I,K]*B[K]+S;
 MATVEC:=S
•END•;

•PROCEDURE•RCYCLE (R, XR, UR, RAXR, AAXR);
```

```
•VALUE•R;
•REAL•R, RAXR, AAXR;
•REAL• •ARRAY•XR, UR;
•BEGIN•
•INTEGER•ITERCNT, COUNTER, RESET;
•REAL•PRXR, GRADL, DISTANCE;
•REAL• •ARRAY•GRAD, DGRAD, DIR, SIGMA, YVEC[1:N], H[1:N,1:N],
  GXR[1:M];
•SWITCH•SWR:=RESTART, ENDPOINT, LAST;
•REAL• •PROCEDURE•PENALTY (P, T, Q, GT, REJECT);
•VALUE•T;  •REAL•T;
•INTEGER•REJECT;
•REAL• •ARRAY•P, Q, GT;
•BEGIN•
•REAL•BARRIER, LOSS, PEN;
•REAL• •ARRAY•XT[1:N];
•SWITCH•SWP:=FIN;
 PENALTY:=PEN:=LOSS:=0.0;
 REJECT:=0;
 BARRIER:=1.0;
•FOR•J:=1•STEP•1•UNTIL•N•DO•
 XT[J]:=P[J]+T*Q[J];
•IF•NONNEG•THEN•
•BEGIN•
•FOR•J:=1•STEP•1•UNTIL•N•DO•
•IF•XTYPE[J]=2•THEN•
•BEGIN•
•IF•LOGP[M+J]•THEN•
•BEGIN•
•IF•XT[J]•GR•10-10•THEN•
 BARRIER:=BARRIER*XT[J]•ELSE•
•BEGIN•
 REJECT:=M+J;
 NRP:=NRP+1;
•GOTO•FIN
•END•
•END• •ELSE•
•IF•XT[J]•LS•0.0•THEN•
 LOSS:=LOSS+XT[J]↑2;
•END•
```

```
·END·;
 FUNCTIONS (XT, GT);
 NC2: =NC2+1;
·FOR·I: =1·STEP·1·UNTIL·M·DO·
·BEGIN·
·IF·GTYPE[I]=1·THEN·
 PEN: =PEN+GT[I]·ELSE·
·BEGIN·
·IF·GTYPE[I]=2·THEN·
·BEGIN·
·IF·LOGP[I]·THEN·
·BEGIN·
·IF·GT[I]·GR·10-10·THEN·
 BARRIER: =BARRIER*GT[I]·ELSE·
·BEGIN·
 REJECT: =I;
 NRP: =NRP+1;
·GOTO·FIN
·END·
·END· ·ELSE·
·IF·GT[I]·LS·0. 0·THEN·LOSS: =LOSS+GT[I]↑2
·END· ·ELSE.
·IF·GTYPE[I]=3·THEN·
 LOSS: =LOSS+GT[I]↑2
·END·
·END·;
 PENALTY: =PEN-R*LN (BARRIER)+LOSS/R;
 FIN:
·END·;

·PROCEDURE·GRADIENT (XT, GI, DPIDX, ANALYTIC);
·VALUE·ANALYTIC; ·BOOLEAN·ANALYTIC;
·REAL··ARRAY·XT, GT, DPIDX;
·BEGIN·
·REAL·HJ, DJ, XTJ, DPJ;
·INTEGER·UPPER;
·REAL··ARRAY·GDELTA, DGDXJ [1:M], DIFF [1:M, 1:2], DGDX [1:M,
   1:N];
 UPPER: = ·IF·IWO ·THEN·1·ELSE·2;
·IF·ANALYTIC·THEN·
```

```
• BEGIN •
• F⊘R • J: =1 • STEP • 1 • UNTIL • N • D⊘ •
• F⊘R • I: =1 • STEP • 1 • UNTIL • M • D⊘ •
 DGDX[I, J]: = 0. 0;
 DERIVATIVES (XT, DGDX);
 NC3: =NC3+1;
• END •;
• F⊘R • J: =1 • STEP • 1 • UNTIL • N • D⊘ •
• IF • XTYPE[J]= 0 • THEN •
 DPTDX[J]: = 0. 0 • ELSE •
• BEGIN •
 XTJ: =XT[J];
• IF • ANALYTIC • THEN •
• BEGIN •
• F⊘R • I: =1 • STEP • 1 • UNTIL • M • D⊘ •
 DGDXJ[I]: =DGDX[I, J]
• END • • ELSE •
• BEGIN •
 HJ: = 10-2*ABS(XTJ) + 10-5;
• F⊘R • K: =1 • STEP • 1 • UNTIL • UPPER • D⊘ •
• BEGIN •
 DJ: =HJ/K;
 NC1: =NC1+2;
 XT[J]: =XTJ+DJ;
 FUNCTI⊘NS (XT, GDELTA);
• F⊘R • I: =1 • STEP • 1 • UNTIL • M • D⊘ •
 DIFF[I, K]: =GDELTA[I];
 XT[J]: =XTJ-DJ;
 FUNCTI⊘NS (XT, GDELTA);
• F⊘R • I: =1 • STEP • 1 • UNTIL • M • D⊘ •
 DIFF[I, K]: = (DIFF[I, K]-GDELTA[I])/(2. 0*DJ)
• END •;
 XT[J]: =XTJ;
• F⊘R • I: =1 • STEP • 1 • UNTIL • M • D⊘ •
 DGDXJ[I]: = • IF • TW⊘ • THEN •
 DIFF[I, 1] • ELSE •
 (4. 0*DIFF[I, 2]-DIFF[I, 1])/3. 0;
• END •;
 DPJ: = 0. 0;
• F⊘R • I: =1 • STEP • 1 • UNTIL • M • D⊘ •
```

```
•BEGIN•
•IF•GTYPE[I]=1•THEN•
DPJ:=DPJ+DGDXJ[I]•ELSE•
•BEGIN•
•IF•GTYPE[I]=2•THEN•
•BEGIN•
•IF•LOGP[I]•THEN•
DPJ:=DPJ-R*DGDXJ[I]/GT[I]•ELSE•
•IF•GT[I]•LS•0.0•THEN•
DPJ:=DPJ+2.0*DGDXJ[I]*GT[I]/R
•END••ELSE•
•IF•GTYPE[I]=3•THEN•
DPJ:=DPJ+2.0*DGDXJ[I]*GT[I]/R
•END•
•END•;
•IF•XTYPE[J]=2•THEN•
•BEGIN•
•IF•LOGP[M+J]•THEN•
DPJ:=DPJ-R/XTJ•ELSE•
•IF•XTJ•LS•0.0•THEN•
DPJ:=DPJ+2.0*XTJ/R•
•END•;
DPTDX[J]:=DPJ
•END•
•END•;

•PROCEDURE•LINEMIN(POINT, PRPOINT, GPOINT, GRADPR, S, DIF-
  FP, DIFFG, RACC, AACC);
•VALUE•RACC, AACC;
•REAL•PRPOINT, RACC, AACC;
•REAL••ARRAY•POINT, GPOINT, GRADPR, S, DIFFP, DIFFG;
•BEGIN•
•INTEGER•NSTEPS, IDLE;
•REAL•LAMBDA, PRA, PRB, PRC, PRD, A, B, C, D, E, MULTIPLIER,
  DESCENT, PA, PC;
•REAL••ARRAY•GA, GB, GC, GD[1:M];
•SWITCH•SWL:=FORWARD, INTER1, INTER2, LAST;
•BOOLEAN••PROCEDURE•READY (NEW, OLD);
•VALUE•NEW, OLD; •REAL•NEW, OLD;
•BEGIN•
```

```
•REAL•SJ, R1, A1, TERM, •SWITCH•SWR:=L, TERM:=ABS (OLD-
  NEW),
 READY:=•TRUE•;
•IF•TERM•LS•10-70•THEN••BEGIN•R1:=A1:=1050,•GOTO•L•END•,
 R1:=RACC/TERM, A1:=AACC/TERM,
 L:•FOR•I:=1•STEP•1•UNTIL•N•DO•
•IF•XTYPE[I]•NQ•0•THEN•
•BEGIN•
 SJ:=S[J];
•IF•ABS(SJ)•GR•R1*ABS (POINT[J]+NEW*SJ)+A1•THEN•
•BEGIN•
 READY:=•FALSE•,
 J:=N
•END•
•END•
•END•,

•PROCEDURE•SHIFT1 (Y, Z, PRY, PRZ, GY, GZ);
•VALUE•Z, PRZ; •REAL•Y, Z, PRY, PRZ,
•REAL••ARRAY•GY, GZ,
•BEGIN•
 Y:=Z;
 PRY:=PRZ,
•FOR•I:=1•STEP•1•UNTIL•M•DO•
 GY[I]:=GZ[I]
•END•,

 NSEARCH:=NSEARCH+1;
 DESCENT:=VECVEC (1,N,0,GRADPR, S),
 MULTIPLIER:=2. 0;
 LAMBDA:=ABS (1. 0/DESCENT),
•IF•LAMBDA•GR•1. 0•THEN•LAMBDA:=1. 0;
 A:=B:=C:=0. 0,
 PRA:=PRB:=PRC:=PRPOINT;
•FOR•I:=1•STEP•1•UNTIL•M•DO•
 GA[I]:=GB[I]:=GC[I]:=GPOINT[I],
 FORWARD:•FOR•NSTEPS:=1•STEP•1•UNTIL•50•DO•
•BEGIN•
 C:=B+LAMBDA;
 PRC:=PENALTY (POINT, C, S, GC, IDLE),
```

```
• IF • IDLE • GR • 0 • THEN•
• BEGIN•
  LAMBDA: = 0. 5*LAMBDA,
  MULTIPLIER: = 0. 5
• END••ELSE•
• BEGIN•
• IF • PRC • LQ • PRB • THEN•
• BEGIN•
  LAMBDA: = MULTIPLIER*LAMBDA,
  SHIFT1 (A, B, PRA, PRB, GA, GB),
  SHIFT1 (B, C, PRB, PRC, GB, GC)
• END••ELSE•
• GOTO • INTER1
• END•
• END•;

  INTER1: • IF • A = 0. 0 • AND • B = 0. 0 • AND • C • GR • 0. 0 • AND •
    DESCENT • LS • 0. 0 • AND • PRB • LS • PRC • THEN •
• BEGIN•
• FOR • NSTEPS: = 1 • STEP • 1 • UNTIL • 50 • DO•
• BEGIN•
  D: = -(0. 5*DESCENT*C↑2)/(PRC-PRB-DESCENT*C),
  PRD: = PENALTY (POINT, D, S, GD, IDLE),
  IF • IDLE • GR • 0 • THEN •
• BEGIN•
  LAMBDA: = 0. 5*D;
  MULTIPLIER: = 0. 5;
• GOTO • FORWARD
• END•;

• IF • PRD • GR • PRB • THEN•
• BEGIN•
• IF • READY (C, 0. 0) • THEN••GOTO • LAST,
  SHIFT 1 (C, D, PRC, PRD, GC, GD)
• END••ELSE •
• BEGIN•
  SHIFT 1 (B, D, PRB, PRD, GB, GD),
• GOTO • INTER2
• END•
• END•
```

```
•END•,

INTER2: •IF • A •LS• B •AND• B •LS• C •AND• PRA•GQ•PRB•AND•
  PRB•LQ•PRC•THEN•
•BEGIN•
•FOR•NSTEPS:=1•STEP•1•UNTIL•50•DO•
•BEGIN•
PC:=(A-B)*(PRC-PRB),
•IF•PC=0.0•THEN•
D:=0.5*(B+C)•ELSE•
•BEGIN•
PA:=(B-C)*(PRA-PRB),
D:=0.5*((A+B)*PC+(B+C)*PA)/(PA+PC)
•END•,

E:=B;
PRD:=PENALTY(POINT, D, S, GD, IDLE),
•IF•IDLE•GR•0•THEN•
•BEGIN•
•IF•D•GR•B•THEN•
LAMBDA:=0.5*(D-B)•ELSE•
•EGIN•
LAMBDA:=0.5*(D-A),
SHIFT1(B, A, PRB, PRA, GB, GA),
SHIFT1(A, 0.0, PRA, PRPOINT, GA, GPOINT)
•END•,
MULTIPLIER:=0.5;
•GOTO•FORWARD
•END•,
•IF•D•LS•B•THEN•
•BEGIN•
•IF•PRD•LS•PRB•THEN•
•BEGIN•
SHIFT1(C, B, PRC, PRB, GC, GB),
SHIFT1(B, D, PRB, PRD, GB, GD)
•END••ELSE•
SHIFT1(A, D, PRA, PRD, GA, GD)
•END••ELSE•
•BEGIN•
•IF•PRD•LS•PRB•THEN•
```

```
·BEGIN·
 SHIFT1(A, B, PRA, PRB, GA, GB);
 SHIFT1(B, D, PRB, PRD, GB, GD)
·END··ELSE·
 SHIFT1(C, D, PRC, PRD, GC, GD)
·END·;
·IF·READY(B, E)·THEN··GOTO·LAST
·END·
·END·;
 LAST:PRPOINT:=PRB;
·FOR·J:=1·STEP·1·UNTIL·N·DO·
·BEGIN·
 DIFFP[J]:=B*S[J];
 DIFFG[J]:=GRADPR[J];
 POINT[J]:=POINT[J]+DIFFP[J]
·END·;
·FOR·I:=1·STEP·1·UNTIL·M·DO·GPOINT[I]:=GB[I];
 GRADIENT(POINT, GPOINT, GRADPR, ANALYTIC);
·FOR·J:=1·STEP·1·UNTIL·N·DO·
 DIFFG[J]:=GRADPR[J]-DIFFG[J]
·END·;

·PROCEDURE·OUTPUT1(CYCLE, R, ITERCNT, IEFF, ANALYTIC,
   CONVERGED, NLP, HEADING, PRXR, GRADL, DISTANCE, XR,
   GRAD, DGRAD, GXR, UR, N, M, COC);
·REAL·R, PRXR, GRADL, DISTANCE;
·INTEGER·CYCLE, ITERCNT, IEFF, N, M, COC;
·BOOLEAN·ANALYTIC, CONVERGED, NLP, HEADING;
·REAL··ARRAY·XR, GRAD, DGRAD, GXR, UR;
·BEGIN·
·INTEGER·I, J;
·SWITCH·SWO:=OMIT;
·IF·LINAGE=0·THEN·LINEFEED(3);
·IF·ITERCNT=0·AND·NLP·THEN·
·BEGIN·
 LINEFEED(2);
 PRINTTEXT(30, ·(BEGIN·), 1, ·(OF·), 1, ·(CYCLE·), 1, ·(·), 0);
 PUTVI(CYCLE, 3);
 PRINTTEXT(2, ·(FOR·), 1, ·(R·), 1, ·(EQUAL·), 1, ·(TO·), 1);
 PUTR(R), LINEFEED(1);
```

```
·END·;
·IF·COC=3·THEN··GOTO·OMIT;
·IF·HEADING·THEN·
·BEGIN·
 LINEFEED(2);
 SPACE(30); PUTS(·(ITERATION·));
 PRINTTEXT(24, ·(PENALTY·), 1, ·(VALUE·), 17, ·(GRADIENT·),
   1, ·(LENGTH·), 22);
 PUTS(·(DISTANCE·)); ; LINEFEED(1)
·END·;
 LINEFEED(1);
 PUTVI(IEFF, 36);
 SPACE(24);
 PUTR(PRXR);
 SPACE(11); PUTR(GRADL);
 SPACE(14); PUTR(DISTANCE); LINEFEED(1);
 HEADING:=·FALSE·;
 OMIT:·IF·IEFF=0·OR·CONVERGED·OR·COC=2·THEN·
·BEGIN·
 LINEFEED(2);
 PRINTTEXT(31, ·(VARIABLE·), 23, ·(SOLUTION·), 1, ·(VECTOR·),
   14, ·(·), 0);
 PRINTTEXT (1, ·(GRADIENT·), 1, ·(OF·), 1, ·(PEN·F··),
   16, ·(·), 0);
·IF·ANALYTIC·AND·IEFF=0·THEN·PUTS(·(GRAD[DIFFEREN-
   CES]·)); LINEFEED(1);
·FOR·J:=1·STEP·1·UNTIL·N·DO·
·BEGIN·
 LINEFEED(1);
 SPACE(31);
 PUTS(·(X·));
 PUTVI(J, 7);
 SPACE(21);
 PUTR(XR[J]);
 SPACE(11);
 PUTR(GRAD[J]);
 SPACE(14);
·IF·ANALYTIC·AND·IEFF=0·THEN·PUTR(DGRAD[J]);
 LINEFEED(1)
·END·;
```

```
 HEADING: = •TRUE•
•END•;
•IF•IEFF = 0 •OR•CONVERGED•THEN•
•BEGIN•
 LINEFEED(2);
 PRINTTEXT(31, •(FUNCTION•), 23, •(FUNCTION•), 1, •(VALUES•),
   0, •(•), 0);
•IF•NLP•AND•CONVERGED•THEN•
 PRINTTEXT(17, •(DUAL•), 1, •(SOLUTION•), 0, •(•), 0, •(•), 0);
   LINEFEED(1);
•FOR•I: = 1•STEP•1•UNTIL•M•DO•
•BEGIN•
 LINEFEED(1);
 SPACE(31); PUTS(•(GX•));
 PUTVI(I, 6);
 SPACE(21);
 PUTR(GXR[I]);
 SPACE(11);
•IF•NLP•AND•CONVERGED•THEN•PUTR(UR[I]); LINEFEED(1);
•END•;
 HEADING: = •TRUE•
•END•;
•IF•NLP•AND•CONVERGED•THEN•
•BEGIN•
 LINEFEED(2);
 PRINTTEXT(30, •(END•), 1, •(OF•), 1, •(CYCLE•), 1, •(•), 0);
 PUTVI(CYCLE, 3);
 PRINTTEXT(2, .(FOR•), 1, .(R•), 1, .(EQUAL•), 1, .(TO•), 1);
 PUTR(R); LINEFEED(1)
•END•
•END•;
 ITERCNT: =COUNTER: = 0;
 DISTANCE: =0. 0;
 CONVERGED: = •FALSE•;
•FOR•J: = 1•STEP•1•UNTIL•N•DO•
 GRAD[J]: =DIR[J]: =SIGMA[J]: =YVEC[J]: = 0. 0;
•FOR•I: = 1•STEP•1•UNTIL•M•DO•GXR[I]: =UR[I]: = 0. 0;
 RESTART: PRXR: =PENALTY(XR, 0. 0, XR, GXR, RESET);
•IF•RESET•GR•0•THEN•
•BEGIN•
```

```
LOGP[RESET]:=.FALSE.;
.GOTO.RESTART
.END.;
GRADIENT(XR, GXR, GRAD, ANALYTIC);
.IF.ANALYTIC.AND.IEFF=0.THEN.
GRADIENT(XR, GXR, DGRAD, .FALSE.);
GRADL:=SQRT(VECVEC(1, N, 0, GRAD, GRAD));
.FOR.ITERCNT:=0.STEP.1.UNTIL.COUNTER+NV.DO.
.BEGIN.
.INTEGER.TERM; TERM:=2*NV; TERM:=(ITERCNT//TERM)*TERM;
.IF.(ITERCNT=0).OR.(ITERCNT=TERM.AND.GRADL.GR.1.0).
.THEN.
.BEGIN.
.FOR.I:=1.STEP.1.UNTIL.N.DO.
.FOR.K:=1.STEP.1.UNTIL.N.DO.
H[I, K]:=.IF.I=K.THEN.1.0.ELSE.0.0
.END. .ELSE.
.BEGIN.
.REAL.SI, HI, SIGMAY, YHY;
.REAL. .ARRAY.HY[1:N];
.FOR.I:=1.STEP.1.UNTIL.N.DO.
HY[I]:=MATVEC(1, N, I, H, YVEC);
SIGMAY:=VECVEC(1, N, 0, SIGMA, YVEC);
YHY:=VECVEC(1, N, 0, YVEC, HY);
.FOR.I:=1.STEP.1.UNTIL.N.DO.
.BEGIN.
HI:=HY[I]/YHY;
.FOR.K:=1.STEP.1.UNTIL.I.DO.
H[K, I]:=H[I, K]:=H[I, K]+SIGMA[I]*SIGMA[K]/SIGMAY-HI*HY
  [K];
.END.
.END.;
.FOR.J:=1.STEP.1.UNTIL.N.DO.DIR[J]:=MATVEC(1,N, J,H,
  GRAD);
.IF.VECVEC(1, N, 0, GRAD, DIR).GR.0.0.THEN.
.FOR.J:=1.STEP.1.UNTIL.N.DO.DIR[J]:=-DIR[J];
.IF.COC.NQ.0.THEN.OUTPUT1(CYCLE, R, ITERCNT, IEFF, ANAL-
  YTIC, CONVERGED, NLP, HEADING, PRXR, GRADL, DISTANCE,
  XR, GRAD, DGRAD, GXR, CR, N, M, COC);
IEFF:=IEFF+1;
```

```
•IF•IEFF=IMAX•THEN• •GOTO•LAST;
 LINEMIN(XR, PRXR, GXR, GRAD, DIR, SIGMA, YVEC, RAXR, AAXR);
 GRADL:=SQRT(VECVEC(1, N, 0, GRAD, GRAD));
 DISTANCE:=SQRT(VECVEC(1, N, 0, SIGMA, SIGMA));
•IF•DISTANCE=0. 0•THEN• •GOTO•ENDPOINT;
•FOR•J:=1 •STEP•1•UNTIL•N•DO•
•IF•XTYPE[J]•NQ•0•THEN•
•BEGIN•
•IF•ABS(SIGMA[J])•GR•RAXR*ABS(XR[J])+AAXR•THEN;
•BEGIN•
 COUNIER:=ITERCNT;
 J:=N
•END•
•END•
•END•;
 ENDPOINT:CONVERGED:=•TRUE•;
•IF•NLP•THEN•
•FOR•I:=1•STEP•1•UNTIL•M•DO;
•BEGIN•
•IF•GTYPE[I]=1•THEN•
 UR[I]:=1.0•ELSE•
•BEGIN•
•IF•GTYPE[I]=2 •THEN•
•BEGIN•
•IF•LOGP[I]•THEN•
 UR[I]:=R/GXR[I]•ELSE•
 UR[I]:=•IF•GXR[I]•LS•0•THEN•-2. 0*GXR[I]/R•ELSE•0. 0;
•END••ELSE•
•IF•GTYPE[I]=3•THEN•UR[I]:=2. 0*GXR[I]/R
•END•
•END•;
•IF•COC•NQ•0•THEN•OUTPUT1 (CYCLE, R, ITERCNT, IEFF, ANAL-
   YTIC, CONVERGED, NLP, HEADING, PRXR, GRADL, DISTANCE, XR,
   GRAD, DGRAD, GXR, UR, N, M, COC);
 LAST:
•END•;
•PROCEDURE•EXTRAPOL(T, ORDER, DIM, NEW, RESULT, CONV);
•INTEGER•ORDER, DIM;
•BOOLEAN•CONV;
•REAL• •ARRAY•T, NEW, RESULT;
```

```
·BEGIN·
·INTEGER·K;
·REAL·BETA;
·REAL· ·ARRAY·AID[1:DIM, 0:ORDER];
 CONV:=·FALSE·;
·FOR·K:=0·STEP·1·UNTIL·ORDER·DO·
·FOR·J:=1·STEP·1·UNTIL·DIM·DO·AID[J, K]:=T[J, K];
·FOR·J:=1·STEP·1·UNTIL·DIM·DO·T[J, 0]:=NEW[J];
·FOR·K:=1·STEP·1·UNTIL·ORDER·DO·
·BEGIN·
 BETA:=1.0/(1.0-FACTOR↑K);
 CONV:=·TRUE·;
·FOR·J:=1·STEP·1·UNTIL·DIM·DO·
 T[J, K]:=BETA*T[J, K-1]+(1.0-BETA)*AID[J, K-1];
·FOR·J:=1·STEP·1·UNTIL·DIM·DO·
·IF·XTYPE[J]·NQ·0·THEN·
·BEGIN·
·IF·ABS(T[J, K]-AID[J, K-1])·GR·RAXMIN*ABS(T[J, K])+AAXMIN,
·THEN·
·BEGIN·
 CONV:=·FALSE·;
 J:=DIM
·END·
·END·;
·IF·CONV·THEN·ORDER:=K
·END·;
·FOR·J:=1·STEP·1·UNTIL·DIM·DO·
 RESULT[J]:=T[J, ORDER]
·END·;

·PROCEDURE·OUTPUT2(NC1, NC2, NC3, NSEARCH, NRP, NV);
·INTEGER·NC1, NC2, NC3, NSEARCH, NRP, NV;
·BEGIN·
 LINEFEED(2);
 PUTVI(NC1, 36);
 PRINTTEXT (1, ·(EVALUATIONS·), 1, ·(OF·), 1, ·(FUNCTIONS·),
   1, ·(FOR·), 0);
 PRINTTEXT(1, ·(NUM··), 1, ·(DIFF··), 0, ·(·), 0, ·(·), 0);
 LINEFEED(2); PUTVI(NC2, 36);
 PRINTTEXT(1, ·(EVALUATIONS·), 1, ·(OF·), 1, ·(FUNCTIONS·),
   1, ·(FOR·), 0);
```

```
PRINTTEXT(1, •(LINE•), 1, •(SEARCHES•), 0, •(•), 0, •(•), 0);
LINEFEED(2); PUTVI(NC3, 36);
PRINTTEXT(1,•(EVALUATIONS•), 1, •(OF•), 1, •(DERIVATIVES•),
  0, •(•), 0);
LINEFEED(2); PUTVI(NC1+NC2+NC3*NV, 36);
PRINTTEXT(1, •(AEQUIVALENT•), 1, •(FUNCTION•), 1, •(EVALUA-
  TIONS•), 0, •(•), 0);
LINEFEED(2); PUTVI(NC2/NSEARCH, 36);
PRINTTEXT(1,•(EVALUATIONS•), 1, •(OF•), •1, •(FUNCTIONS•),
  1, •(FER•), 0);
PRINTTEXT(1, •(LINE•), 1, •(MINIMUM•), 0, •(•), 0, •(•), 0);
LINEFEED(2); PUTVI(NRP, 36);
PRINTTEXT(1, •(REJECTED•), 1, •(POINTS•), 1, •(BECAUSE•),
  1, •(OF•), 0);
PRINTTEXT(1, •(CONSTRAINT•), 1, •(VIOLATION•), 0, •(•), 0. •
  (•), 0); LINEFEED(1);
•END•;

•PROCEDURE• OUTPUT3(CYCLE, ORDER, N, M, XMIN, UMIN,FUNCTI-
  ONS, HEADING);
•INTEGER•CYCLE, ORDER, N, M;
•BOOLEAN•HEADING;
•REAL• •ARRAY•XMIN, UMIN;
•PROCEDURE•FUNCTIONS;
•BEGIN•
•INTEGER•I, J;
•REAL• •ARRAY•GMIN[1:M];
LINEFEED(2);
PRINTTEXT (30, •(EXTRAPOLATION,•), 1, •(CYCLE•), 1, •(•),
  0, •(•), 0);
PUTVI(CYCLE, 3);
PRINTTEXT(1, •(•), 1, •(ORDER•), 1, •(•), 0, •(•), 0);
PUTVI(ORDER, 3);
LINEFEED(2);
PRINTTEXT(31, •(VARIABLE•), 23, •(SOLUTION•), 1,•(VECTOR•),
  0, •(•), 0);
LINEFEED(1);
•FOR•J:=1•STEP•1•UNTIL•N•DO•
•BEGIN•
LINEFEED(1);
SPACE(31); PUTS(•(X•));
```

```
PUTVI(I, 7);
SPACE(21);
PUTR(XMIN[ J ]); LINEFEED(1)
•END•;
LINEFEED(2);
PRINTTEXT(31, •(FUNCTION•), 23, •(FUNCTION•), 1,•(VALUES•),
  0, •(•), 0);
PRINTTEXT(17, •(DUAL•), 1, •(SOLUTION•), 0, •(•), 0, •(•),
  0); LINEFEED(1);
FUNCTIONS(XMIN, GMIN);
•FOR•I:=1•STEP•1•UNTIL•M•DO•
•BEGIN•
LINEFEED(1);
SPACE(31); PUTS(•(GX•));
PUTVI(I, 6);
SPACE(21);
PUTR(GMIN[ I ]);
SPACE(11);
PUTR(UMIN[ I ]); LINEFEED(1)
•END•;
HEADING:= •TRUE•
END•;
LINAGE:=NV:=IEFF:=NC1:=NC2:=NC3:=NSEARCH:=NRP:=0;
HEADING:=•TRUE•;
MAXCYCLE:= 0;
OBJECTIVE:=CONSTRAINTS:=NLP:=NONNEG:=•FALSE•;
INITR:=1. 0;
FACTOR:= 0.4641588832;
•FOR•I:=1•STEP•1•UNTIL•MPLUSN•DO•LOGP[ I ]:=•FALSE•;
•FOR•J:=1•STEP•1•UNTIL•N•DO•XRMIN[ J ]:=XMIN[ J ]:=X[ J ];
•FOR•J:=1•STEP•1•UNTIL•N•DO•
•BEGIN•
•IF•XTYPE[ J ]•LS•0•OR•XTYPE[ J ]•GR•2•THEN••GOTO•TER-
  MINAL;
•IF•XTYPE[ J ]•NQ•0•THEN•NV:=NV+1;
•IF•XTYPE[ J ]=2•THEN•NONNEG:=CONSTRAINTS:=•TRUE•;
•FOR•K:=0•STEP•1•UNTIL•6•DO•XTABLE[ J, K]:=0. 0
•END•;
•FOR•I:=1•STEP•1•UNTIL•M•DO•
•BEGIN•
•IF•GTYPE[ I ]•LS•0•OR•GTYPE[ I ]•GR•3•THEN••GOTO•TER-
```

```
  MINAL;
•IF•GTYPE[I]=1•THEN•ØBJECTIVE:=•TRUE••ELSE•
•IF•GTYPE[I]=2•ØR•GTYPE[I]=3•THEN•CØNSTRAINTS:=
  •TRUE•;
•FØR•K:=0•STEP•1•UNTIL•6•DØ•UTABLE[I,K]:=0.0;
•END•;
•IF•ØBJECTIVE•AND•CØNSTRAINTS•THEN•
•BEGIN•
 INITR:=ABS(ESTIMATE)/100.0;
•IF•INITR•LS•10-2•THEN•INITR:=10-2;
 MAXCYCLE:=30;
 NLP:=•TRUE•;
•FØR•I=1•STEP•1•UNTIL•MPLUSN•DØ•LØGP[I]:=•TRUE•
•END•;
•FØR•CYCLE:=0•STEP•1•UNTIL•MAXCYCLE•DØ•
•BEGIN•
•INTEGER•ØRDERX, ØRDERU;
•BØØLEAN•CØNVX, CØNVU;
 RCYCLE(INITR * FACTØR ↑ CYCLE, XRMIN, URMIN, RAXMIN,
  AAXMIN);
•IF•NØT•CØNVERGED•THEN•
•BEGIN•
•IF•CYCLE=0•THEN•
•FØR•J:=1•STEP•1•UNTIL•N•DØ•XMIN[J]:=XRMIN[J];
•GØTØ•TERMINAL
•END•;
 ØRDERX:=ØRDERU:=•IF•CYCLE•LS•6•THEN•CYCLE•ELSE•6;
 EXTRAPØL(XTABLE, ØRDERX, N, XRMIN, XMIN, CØNVX);
 EXTRAPØL(UTABLE, ØRDERU, M, URMIN, UMIN, CØNVU);
•IF•CØC•NQ•0•THEN•ØUTPUT2(NC1, NC2, NC3, NSEARCH, NRP,
  NV);
•IF•CYCLE•GR•0•THEN•
•BEGIN•
•IF•CØC•NQ•0•THEN•
 ØUTPUT3(CYCLE, ØRDERX, N, M, XMIN, UMIN, FUNCTIØNS, HEADI-
  NG);
•IF•CØNVX•THEN••GØTØ•TERMINAL•ELSE•CØNVERGED:=
  •FALSE•
•END•
•END•;
 TERMINAL:•FØR•J:=1•STEP•1•UNTIL•N•DØ•
```

```
X[I]:=XMIN[I]
·END·;

ANALYTIC:=·TRUE·;
TWO:=·TRUE·;
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
·FOR·I:=1, 2, 3·DO·X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
·FOR·I:=1·STEP·1·UNTIL·6·DO·XTYPE[I]:=1;
·FOR·I:=1, 2, 3, 4·DO·GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
·END·
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR'I:=1, 2, 3'DO'X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR'I:=1'STEP'1'UNTIL'6'DO'XTYPE[I]:=1;
'FOR'I:=1, 2, 3, 4'DO'GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[ I ]:=XMIN[ I ]
•END•;

ANALYTIC:=•TRUE•;
TWO:=•TRUE•;
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
•FOR•I:=1, 2, 3•DO•X[ I ]:=1; X[ 4 ]:=3; X[ 5 ]:=0; X[ 6 ]:=5;
•FOR•I:=1•STEP•1•UNTIL•6•DO•XTYPE[ I ]:=1;
•FOR•I:=1, 2, 3, 4•DO•GTYPE[ I ]:=2, GTYPE[ 5 ]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
•END•
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR'I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR'I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR'I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
·END·;

ANALYTIC:=·TRUE·;
TWO:=·TRUE·;
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
·FOR·I:=1, 2, 3·DO·X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
·FOR·I:=1·STEP·1·UNTIL·6·DO·XTYPE[I]:=1;
·FOR·I:=1, 2, 3, 4·DO·GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
·END·
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[ I ]: = XMIN[ I ]
'END';

ANALYTIC: = 'TRUE';
TWO: = 'TRUE';
IMAX: = 100;
COC: = 1;
ESTIMATE: = 5;
RAXMIN: = AAXMIN: = 10-5;
N: = 6;
M: = 5;
MPLUSN: = N + M;
'FOR' I: = 1, 2, 3 'DO' X[ I ]: = 1; X[ 4 ]: = 3; X[ 5 ]: = 0; X[ 6 ]: = 5;
'FOR' I: = 1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[ I ]: = 1;
'FOR' I: = 1, 2, 3, 4 'DO' GTYPE[ I ]: = 2; GTYPE[ 5 ]: = 1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
·END·;

ANALYTIC:=·TRUE·;
TWO:=·TRUE·;
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
·FOR·I:=1, 2, 3·DO·X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
·FOR·I:=1·STEP·1·UNTIL·6·DO·XTYPE[I]:=1;
·FOR·I:=1, 2, 3, 4·DO·GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
·END·
```

```
X[I]:=XMIN[I]
.END.;

ANALYTIC:=.TRUE.;
TWO:=.TRUE.;
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
.FOR.I:=1, 2, 3.DO.X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
.FOR.I:=1.STEP.1.UNTIL.6.DO.XTYPE[I]:=1;
.FOR.I:=1, 2, 3, 4.DO.GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
.END.
```

906290

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=₁₀-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1,2,3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1,2,3,4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR'I:=1,2,3'DO'X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR'I:=1'STEP'1'UNTIL'6'DO'XTYPE[I]:=1;
'FOR'I:=1,2,3,4'DO'GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
 X[I]:=XMIN[I]
'END';

 ANALYTIC:='TRUE';
 TWO:='TRUE';
 IMAX:=100;
 COC:=1;
 ESTIMATE:=5;
 RAXMIN:=AAXMIN:=10-5;
 N:=6;
 M:=5;
 MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
 MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
   N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
   IMAX);
'END'
```

```
X[I]:=XMIN[I]
.END.;

ANALYTIC:=.TRUE.;
TWØ:=.TRUE.;
IMAX:=100;
CØC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
.FØR.I:=1, 2, 3.DØ.X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
.FØR.I:=1.STEP.1.UNTIL.6.DØ.XTYPE[I]:=1;
.FØR.I:=1, 2, 3, 4.DØ.GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIØNS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWØ, RAXMIN, AAXMIN, ESTIMATE, CØNVERGED, CØC,
  IMAX);
.END.
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[ I ]: = XMIN[ I ]
·END·;

ANALYTIC: = ·TRUE·;
TWO: = ·TRUE·;
IMAX: = 100;
COC: = 1;
ESTIMATE: = 5;
RAXMIN: = AAXMIN: = 10-5;
N: = 6;
M: = 5;
MPLUSN: = N + M;
·FOR·I: = 1, 2, 3·DO·X[ I ]: = 1; X[ 4 ]: = 3; X[ 5 ]: = 0; X[ 6 ]: = 5;
·FOR·I: = 1·STEP·1·UNTIL·6·DO·XTYPE[ I ]: = 1;
·FOR·I: = 1, 2, 3, 4·DO·GTYPE[ I ]: = 2; GTYPE[ 5 ]: = 1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
·END·
```

506290

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR'I:=1, 2, 3'DO'X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR'I:=1'STEP'1'UNTIL'6'DO'XTYPE[I]:=1;
'FOR'I:=1, 2, 3, 4'DO'GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[I]:=XMIN[I]
'END';

ANALYTIC:='TRUE';
TWO:='TRUE';
IMAX:=100;
COC:=1;
ESTIMATE:=5;
RAXMIN:=AAXMIN:=10-5;
N:=6;
M:=5;
MPLUSN:=N+M;
'FOR' I:=1, 2, 3 'DO' X[I]:=1; X[4]:=3; X[5]:=0; X[6]:=5;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' XTYPE[I]:=1;
'FOR' I:=1, 2, 3, 4 'DO' GTYPE[I]:=2; GTYPE[5]:=1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
'END'
```

```
X[ I ]: = XMIN[ I ]
·END·;

ANALYTIC: = ·TRUE·;
TWO: = ·TRUE·;
IMAX: = 100;
COC: = 1;
ESTIMATE: = 5;
RAXMIN: = AAXMIN: = 10-5;
N: = 6;
M: = 5;
MPLUSN: = N + M;
·FOR· I: = 1, 2, 3 ·DO· X[ I ]: = 1; X[ 4 ]: = 3; X[ 5 ]: = 0; X[ 6 ]: = 5;
·FOR· I: = 1 ·STEP· 1 ·UNTIL· 6 ·DO· XTYPE[ I ]: = 1;
·FOR· I: = 1, 2, 3, 4 ·DO· GTYPE[ I ]: = 2; GTYPE[ 5 ]: = 1;
MINIMIZE(X, FUNCTIONS, XTYPE, GTYPE, ANALYTIC, DERIVATIVES,
  N, M, TWO, RAXMIN, AAXMIN, ESTIMATE, CONVERGED, COC,
  IMAX);
·END·
```