

# 第四章 无约束优化方法

4-1 概述

4-2 最速下降法（梯度法）

4-3 牛顿型方法

4-4 变尺度法

4-5 无约束极值问题的直接数值解法

4-6 坐标轮换法

4-7 模式搜索法（基本共轭方向法）

4-8 Powell 共轭方向法

4-9 单形替换法

第1章所列举的机械优化设计问题，都是在一定的限制条件下追求某一指标为最小，它们都属于约束优化问题。工程问题大都如此。

### *为什么要研究无约束优化问题?*

- (1) 有些实际问题，其数学模型本身就是一个无约束优化问题。
- (2) 通过熟悉它的解法可以为研究约束优化问题打下良好的基础。
- (3) 约束优化问题的求解可以通过一系列无约束优化方法来达到。所以无约束优化问题的解法是优化设计方法的基本组成部分，也是优化方法的基础。

(4) 对于多维无约束问题来说，古典极值理论中令一阶导数为零，但要求二阶可微，且要判断海赛矩阵为正定才能求得极小点，这种方法有理论意义，但无实用价值。和一维问题一样，若多元函数 $F(\mathbf{X})$ 不可微，亦无法求解。但古典极值理论是无约束优化方法发展的基础。

## 4-1 概述

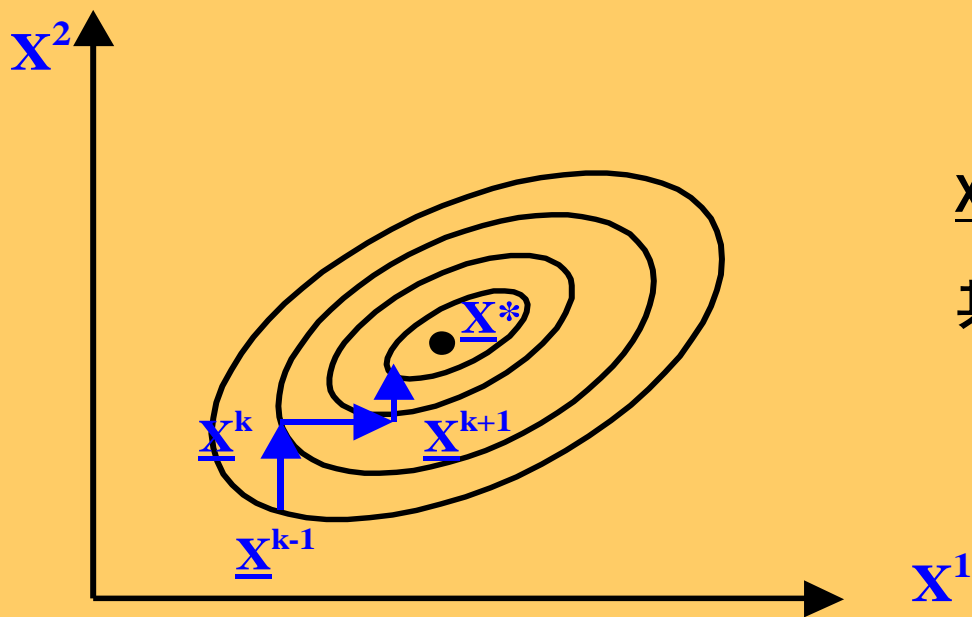
无约束优化问题的数学模型为：

$$\min f(\underline{X}) \quad \underline{X} \in \mathbb{R}^n$$

从理论上说可以用解析法求，即

$$\begin{cases} \frac{\partial f}{\partial x_1} = 0 \\ \frac{\partial f}{\partial x_2} = 0 \\ \dots \\ \frac{\partial f}{\partial x_n} = 0 \end{cases}$$

这是一个含有n个未知量，n个方程的方程组，通常这些方程组都是非线型的，求解起来非常困难



$$\underline{x}^{k+1} = \underline{x}^k + \alpha \underline{d}^k$$

其中：  $\underline{d}^k$  为搜索方向

$\alpha$  为步长

其中对于每一点  $f(\underline{x}^{k+1}) < f(\underline{x}^k)$

无约束优化问题是：

求 $n$ 维设计变量  $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$

使目标函数  $f(\mathbf{x}) \rightarrow \min$

$$\min f(\mathbf{x}) \quad \mathbf{x} \in R^n$$

目前已研究出很多种无约束优化方法，它们的主要不同点在于构造搜索方向上的差别。

按确定方向  $\underline{d}^k$  的方法不同 {

- 解析数值解法：利用导数信息的算法
- 它是通过函数值的一阶或二阶偏导矩阵来确定迭代的方向和步长
- 直接数值解法：不利用导数信息的算法
- 它是通过函数值的计算和比较来确定迭代的方向和步长

无约束  
优化方法 {

- 解析数值解法：梯度法，牛顿法，变尺度法
- 直接数值解法：坐标轮换法，共轭方向法，*Powell*法



## 4-2 最速下降法（梯度法）

# 1 基本思想

迭代过程中的某一点 $\underline{X}^k$ 处，目标函数的负梯度方向 $-\nabla f(\underline{X}^k)$ 是函数的最快下降方向

$$\underline{d}^k = -\nabla f(\underline{X}^k) = -\left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

注意：

(1) 梯度法它通过将搜索方向定为负梯度方向这一性质，就将n维无约束极小化问题转化为一系列沿负梯度方向进行搜索寻优的方法

(2) 这个最速下降法只是在 $\underline{X}^k$ 点的临域内有效，并不一定是全局的最优

## 2 迭代公式

$$\underline{X}^{k+1} = \underline{X}^k - \alpha_k \nabla f(\underline{X}^k)$$

$\alpha_k$  (步长因子):  $\underline{X}^k$  沿负梯度方向的最优步长

可以看到式中,  $\underline{X}^k$ ,  $\nabla f(\underline{X}^k)$  均已知, 要想求得  $\underline{X}^{k+1}$ ,

求  $\alpha_k$  就成为关键

如何求  $\alpha_k$ ?

$$\Phi(\alpha) = f(\underline{X}^{k+1}) = f(\underline{X}^k + \alpha \underline{d}^k) \longrightarrow \min$$

$$\alpha^k = ?$$

即为一维搜索问题, 仅有一个变量

一元函数求极值  $\Phi'(\alpha) = 0$  求得  $\alpha_k$

### 3 实例

求目标函数  $f(\mathbf{x}) = x_1^2 + 25x_2^2$  的极小点。

解： 取初始点  $\mathbf{x}^0 = [2, 2]^T$

则初始点处函数值及梯度分别为

$$f(\mathbf{x}^0) = 104$$

$$\nabla f(\mathbf{x}^0) = \begin{bmatrix} 2x_1 \\ 50x_2 \end{bmatrix}_{\mathbf{x}_0} = \begin{bmatrix} 4 \\ 100 \end{bmatrix}$$

沿负梯度方向进行一维搜索，有

$$\mathbf{x}^1 = \mathbf{x}^0 - \alpha_0 \nabla f(\mathbf{x}^0) = \begin{bmatrix} 2 - 4\alpha_0 \\ 2 - 100\alpha_0 \end{bmatrix}$$

$\alpha_0$  为一维搜索最佳步长，应满足极值必要条件

$$f(\mathbf{x}^1) = \min_{\alpha} \{ (2 - 4\alpha)^2 + 25(2 - 100\alpha)^2 \} = \min_{\alpha} \varphi(\alpha)$$

$$\varphi'(\alpha) = -8(2 - 4\alpha_0) - 5\,000(2 - 100\alpha_0) = 0$$

算出一维搜索最佳步长

$$\alpha_0 = \frac{626}{31\,252} = 0.020\,030\,72$$

第一次迭代设计点位置和函数值

$$\mathbf{x}^1 = \begin{bmatrix} 2 - 4\alpha_0 \\ 2 - 100\alpha_0 \end{bmatrix} = \begin{bmatrix} 1.919\,877 \\ -0.307\,178\,5 \times 10^{-2} \end{bmatrix}$$

$$f(\mathbf{x}^1) = 3.686\,164$$

继续作下去，经10次迭代后，得到最优解

$$f(\mathbf{x}^*) = 0$$

这个问题的目标函数的等值线为一簇椭圆, 迭代点从  $x^0$  走的是一段锯齿形路线, 见图4-3。

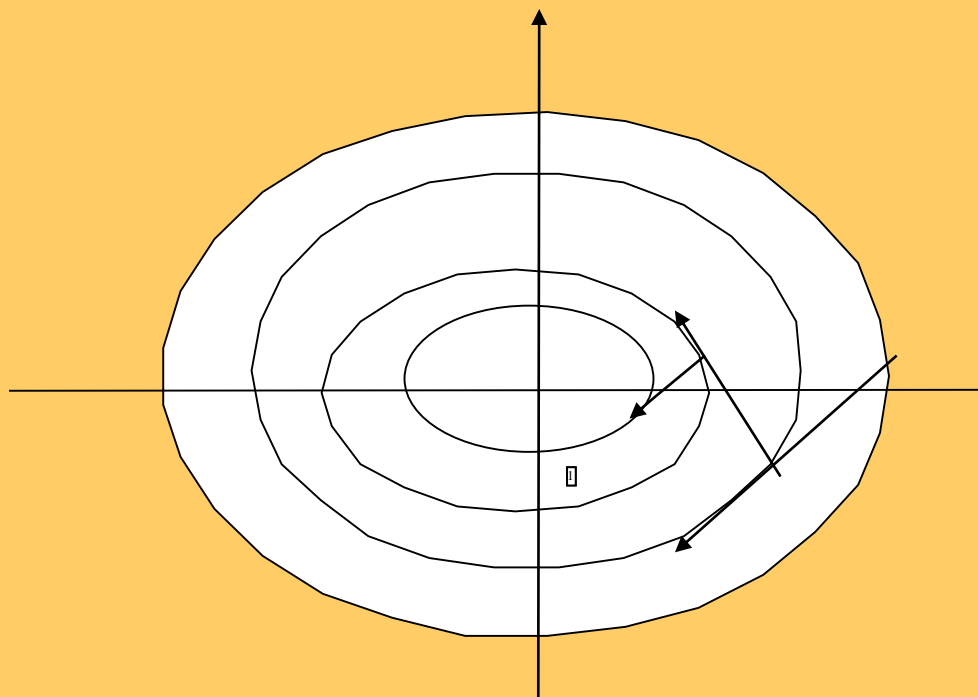


图4-3

将上例中目标函数  $f(\mathbf{x}) = x_1^2 + 25x_2^2$  引入变换

$$y_1 = x_1, \quad y_2 = 5x_2$$

则函数  $f(\mathbf{x})$  变为:  $\phi(y_1, y_2) = y_1^2 + y_2^2$

其等值线由椭圆变成一簇同心圆。

仍从  $\mathbf{x}^0 = [2, 2]^T$  即  $\mathbf{y}^0 = [2, 10]^T$  出发进行最速下降法寻优。此时:

$$\phi(\mathbf{y}^0) = 104$$

$$\nabla \phi(\mathbf{y}^0) = \begin{bmatrix} 2y_1 \\ 2y_2 \end{bmatrix}_{\mathbf{y}^0} = \begin{bmatrix} 4 \\ 20 \end{bmatrix}$$

沿负梯度方向进行一维搜索:

$$\mathbf{y}^1 = \mathbf{y}^0 - \beta_0 \nabla \phi(\mathbf{y}^0)$$

$$\begin{bmatrix} 2 \\ 10 \end{bmatrix} - \beta_0 \begin{bmatrix} 4 \\ 20 \end{bmatrix} = \begin{bmatrix} 2 - 4\beta_0 \\ 10 - 20\beta_0 \end{bmatrix}$$

$\beta$ 为一维搜索最佳步长，可由极值条件：

$$\phi(\mathbf{y}^1) = \min_{\beta} \phi[\mathbf{y}^0 - \beta \nabla \phi(\mathbf{y}^0)] = \min_{\beta} \Phi(\beta)$$

$$\Phi(\beta) = (2 - 4\beta)^2 + (10 - 20\beta)^2$$

$$\text{由 } \Phi'(\beta_0) = 0$$

$$\beta_0 = \frac{26}{52} = 0.5$$

从而算得一步计算后设计点的位置及其目标函数：



$$\mathbf{y}^1 = \begin{bmatrix} 2 - 4\beta_0 \\ 10 - 20\beta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\phi(\mathbf{y}^1) = 0$$

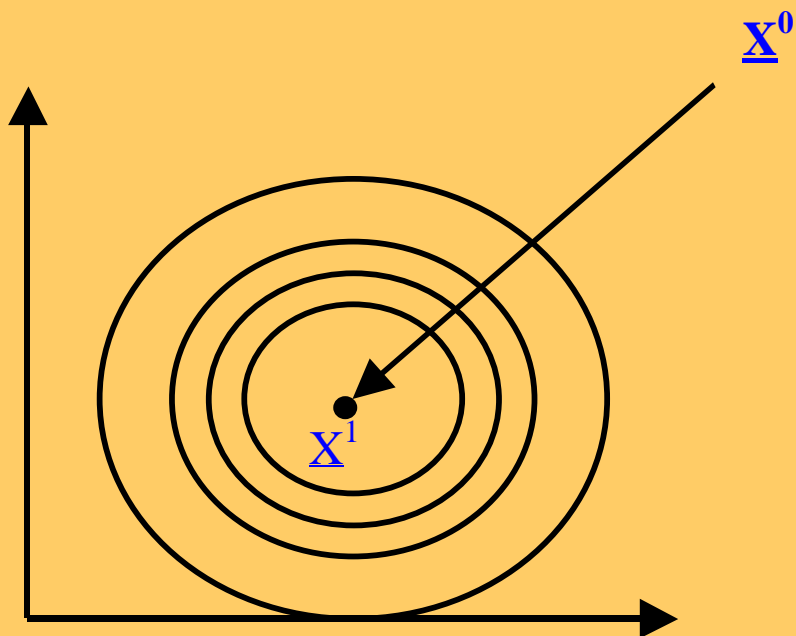
经变换后，只需一次迭代，就可找到最优解。

这是因为经过尺度变换：

$$y_1 = x_1$$

$$y_2 = 5x_2$$

等值线由椭圆变成圆。



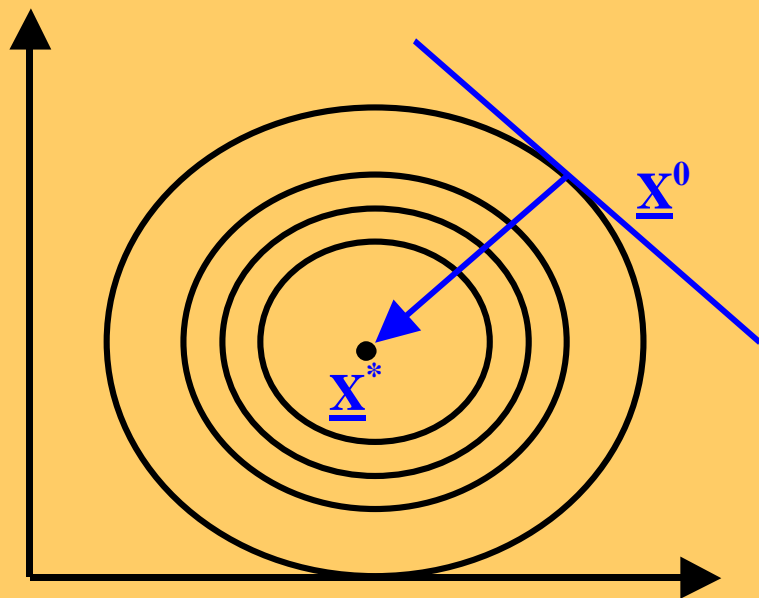
目标函数的等值线是以某点,  
为圆心的一系列同心圆

从  $x^0$  出发, 沿梯度方向一次迭代  
到最小点

实际上当目标函数等值线为同心圆时,  
负梯度方向是可以一次迭代到极值点  
的, 这是梯度的特点之一

## 4 讨论

(1) 若 $f(\underline{x})$ 的等值线为同心圆时，梯度法只需要一次迭代就可以到达最点



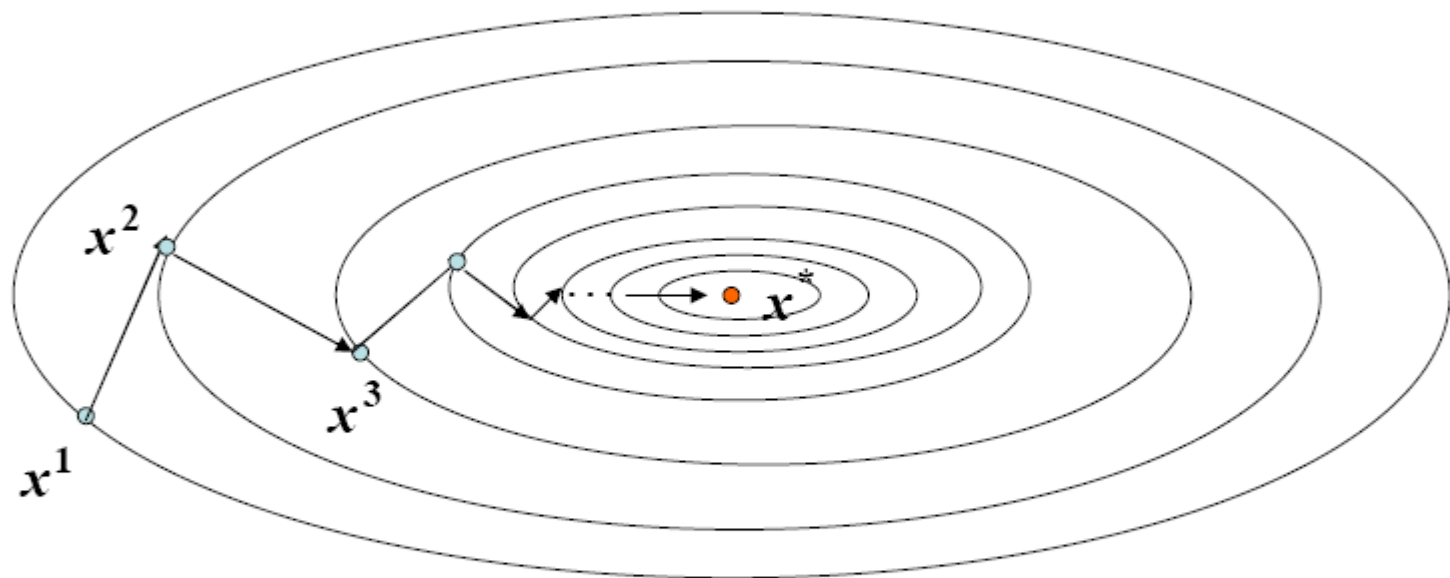
在第二章的对梯度的学习我们知道，梯度方向是和等值线相垂直的方向，或是与等值线在该点的切线方向相垂直的方向

由图中可看到，无论初始点在哪，该点的负梯度方向均指向圆心

在最速下降法中，相邻两个迭代点上的函数梯度相互垂直。而搜索方向就是负梯度方向，因此相邻两个搜索方向互相垂直。这就是说在迭代点向函数极小点靠近的过程，走的是曲折的路线。形成“之”字形的锯齿现象，而且越接近极小点锯齿越细。

## 锯齿现象

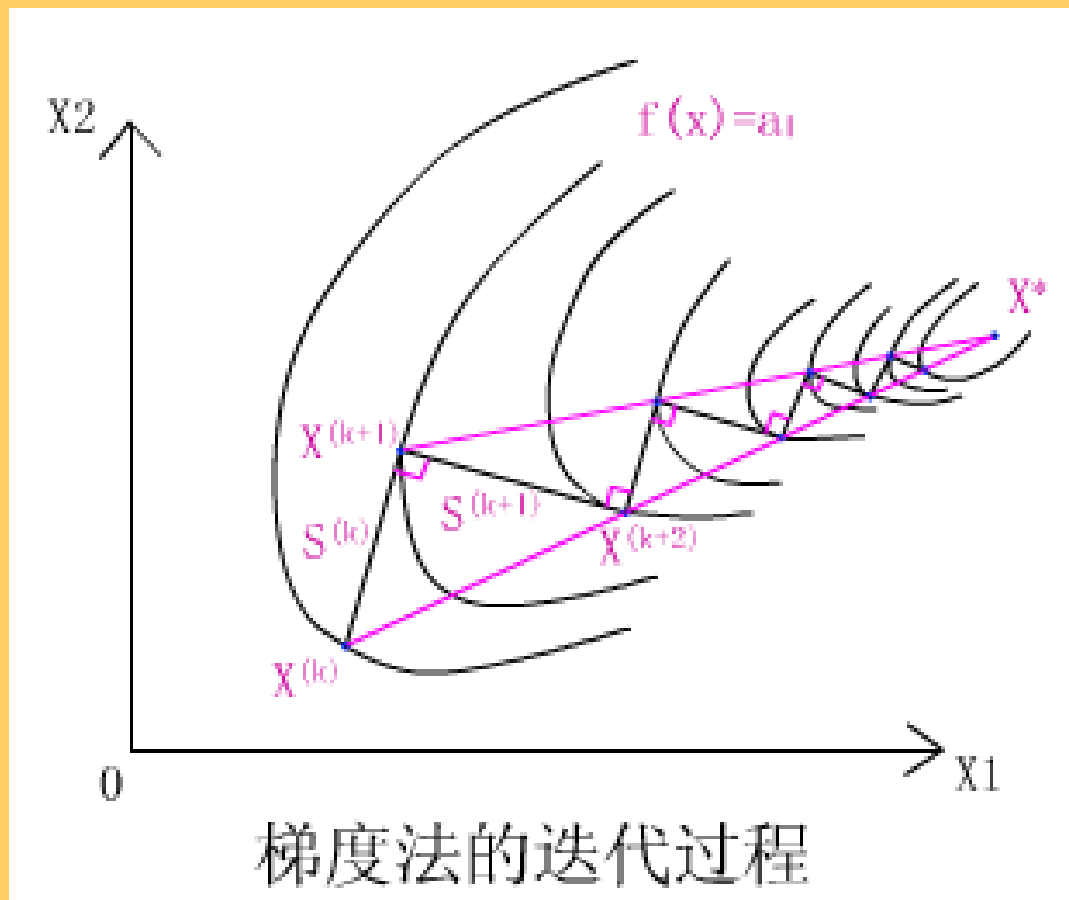
在极小点附近，目标函数可以用二次函数近似，其等值面近似椭球面。



**注** 最速下降方向反映了目标函数的一种局部性质。它只是局部目标函数值下降最快的方向。

最速下降法是线性收敛的算法。

(2) 当 $f(x)$ 的等值线为同心椭圆时，迭代过程走曲线，且在相邻两次时，方向互相垂直



## 分析上图：

(a) 从局部看（即迭代点看），梯度方向是下降最快的方向

(b) 从整体看，梯度方向并不是降最快的方向

当等值线为同心椭圆时，能否一次迭代到达极小点呢？回答是肯定的，此时的迭代方向是牛顿方向

(c) 梯度法的收敛速度和椭圆的长短轴有很大的关系，长短轴相差得越大，就收敛得越慢

当椭圆越扁，收敛的就越慢

(d) 函数下降路径位于两条射线之间，呈锯齿形，越接近中心锯齿越小

(e) 一般而言，越接近极值点，收敛越慢

从特点d) 中可以看出这个特点

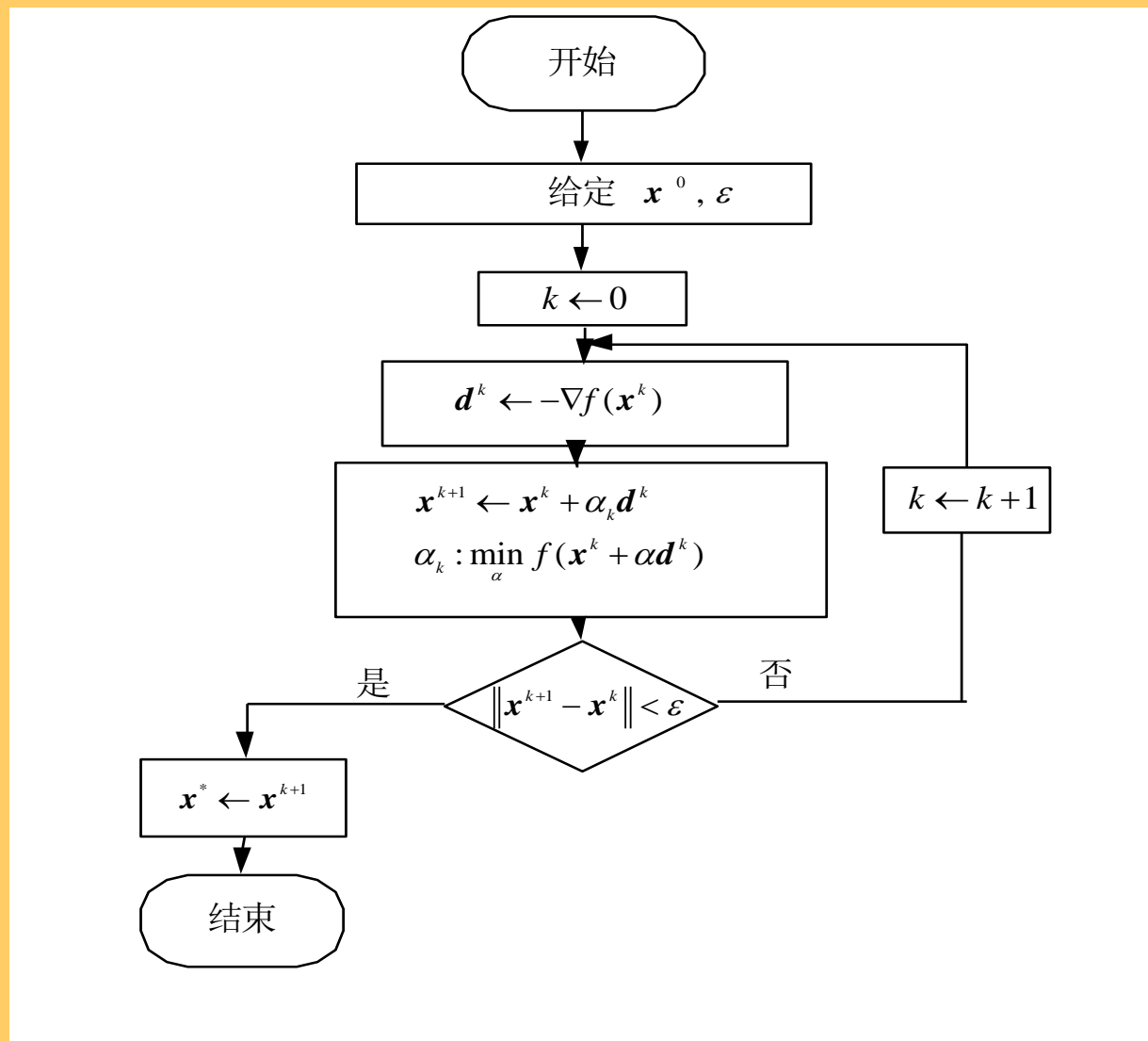
## 方法特点

(1) 初始点可任选，每次迭代计算量小，存储量少，程序简短。即使从一个不好的初始点出发，开始的几步迭代，目标函数值下降很快，然后慢慢逼近局部极小点。

(2) 任意相邻两点的搜索方向是正交的，它的迭代路径为绕道逼近极小点。当迭代点接近极小点时，步长变得很小，越走越慢。



## 5. 最速下降法的程序框图



## 4-3 牛顿型方法

## 4-2 牛顿法及其改进

基本思想：

在 $\mathbf{x}^k$ 邻域内用一个二次函数 $\varphi(\mathbf{x})$ 来近似代替原目标函数，并将 $\varphi(\mathbf{x})$ 的极小点作为对目标函数 $f(\mathbf{x})$ 求优的下一个迭代点 $\mathbf{x}^{k+1}$ 。经多次迭代，使之逼近目标函数 $f(\mathbf{x})$ 的极小点。

牛顿法是求函数极值的最古老算法之一。

$$\begin{aligned} f(\mathbf{x}) \approx \varphi(\mathbf{x}) &= f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T (\mathbf{x} - \mathbf{x}^k) \\ &\quad + \frac{1}{2} (\mathbf{x} - \mathbf{x}^k)^T \nabla^2 f(\mathbf{x}^k) (\mathbf{x} - \mathbf{x}^k) \end{aligned}$$

设 $\mathbf{x}^{k+1}$ 为 $\varphi(\mathbf{x})$ 的极小点

$$\nabla \varphi(\mathbf{x}^{k+1}) = 0$$

$$\nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k) (\mathbf{x}^{k+1} - \mathbf{x}^k) = 0$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots)$$

这就是多元函数求极值的牛顿法迭代公式。

对于二次函数，海赛矩阵 $H$ 是一个常矩阵，其中各元素均为常数。因此，无论从任何点出发，只需一步就可找到极小点。

例4—2 求目标函数  $f(\mathbf{x}) = x_1^2 + 25x_2^2$  的极小点。

解 取初始点  $\mathbf{x}^0 = [2, 2]^T$

$$\mathbf{x}^1 = \mathbf{x}^0 - [\nabla^2 f(\mathbf{x}^0)]^{-1} \nabla f(\mathbf{x}^0) = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{50} \end{bmatrix} \begin{bmatrix} 4 \\ 100 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

经过一次迭代即求得极小点  $\mathbf{x}^* = [0 \ 0]$

函数极小值  $f(\mathbf{x}^*) = 0$

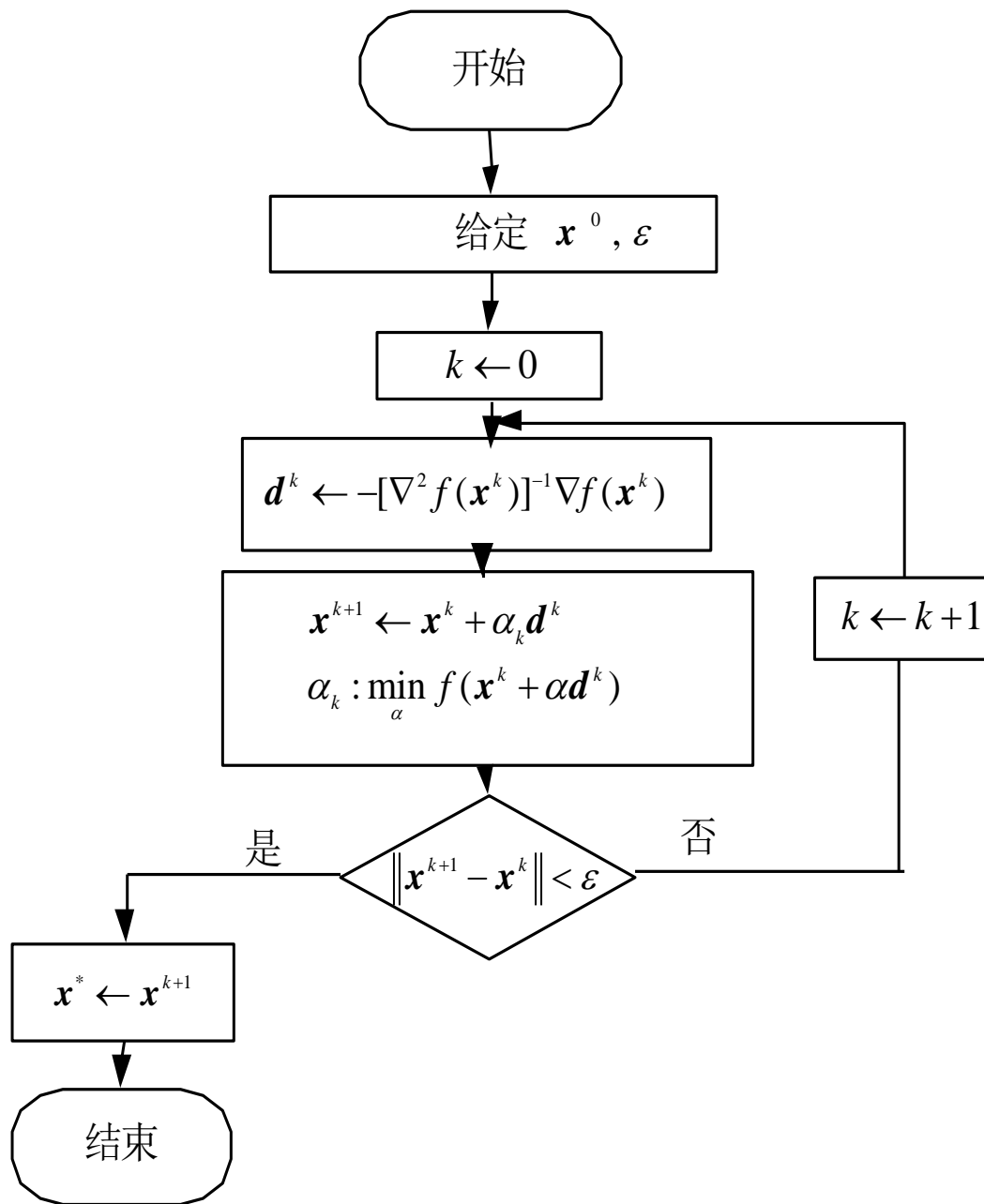
从牛顿法迭代公式的推演中可以看到，迭代点的位置是按照极值条件确定的，其中并未含有沿下降方向搜寻的概念。因此对于非二次函数，如果采用上述牛顿迭代公式，有时会使函数值上升。

### 阻尼牛顿法

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k s^k = \mathbf{x}^k - \alpha_k [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots)$$

$\alpha_k$  阻尼因子，沿牛顿方向进行一维搜索的最佳步长，由下式求得：

$$f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k + \alpha_k s^k) = \min_{\alpha} f(\mathbf{x}^k + \alpha_k s^k)$$



阻尼牛顿法程序框图

### 3. 阻尼牛顿法:

(1) 当 $f(\underline{x})$ 为正定二次函数时, 牛顿方向指向极小点, 迭代一次即可求出极值点

(2) 对于非二次型函数有时会出现 $f(\underline{x}^{k+1}) > f(\underline{x}^k)$ 的现象

(3) 对初始点的 $\underline{x}^0$ 的要求比较高

(4) Hesse矩阵计算困难, 求逆矩阵更加困难

使用有一定的局限性, 由于牛顿法有时会出现函数值上升的情况, 为了克服以上弊病, 提出了“阻尼牛顿法”或“修正牛顿法”

阻尼牛顿法的迭代公式为:

迭代公式: 
$$\underline{x}_{k+1} = \underline{x}_k - \underbrace{\alpha_k}_{\text{阻尼因子}} \underbrace{[H(\underline{x}_k)]^{-1} \nabla f(\underline{x})}_{\text{牛顿方向}}$$

牛顿法和阻尼牛顿法统称为牛顿型方法。这类方法的主要缺点是每次迭代都要计算函数的二阶导数矩阵，并对该矩阵求逆。这样工作量很大。特别是矩阵求逆，当维数高时工作量更大。



# 方法特点

- (1) 初始点应选在 $X^*$ 附近，有一定难度；
- (2) 尽管每次迭代都不会是函数值上升，但不能保证每次下降；
- (3) 若迭代点的海赛矩阵为奇异，则无法求逆矩阵，不能构造牛顿法方向；
- (4) 不仅要计算梯度，还要求海赛矩阵及其逆矩阵，计算量和存储量大。此外，对于二阶不可微的 $F(X)$ 也不适用。

虽然阻尼牛顿法有上述缺点，但在特定条件下它具有收敛最快的优点，并为其他的算法提供了思路 and 理论依据。

## 梯度法与牛顿法:

一般迭代式:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k s^k \quad (k = 0, 1, 2, \dots)$$

梯度法:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots)$$

牛顿法:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots)$$

阻尼牛顿法:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots)$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k A^k \nabla f(\mathbf{x}^k)$$

(5) 梯度方向:

当等值线为圆时,  $-\nabla f(\underline{x})$  指向圆心

当等值线为椭圆时,  $-\nabla f(\underline{x})$  不指向圆心, 呈锯齿状

牛顿方向:

当等值线为椭圆时,  $-[H(\underline{x}_k)]^{-1} \nabla f(\underline{x})$  指向椭圆中心

## 4-4 变尺度法

**DFP变尺度法首先有戴维顿（Davidon）与1959年提出，又于1963年由弗莱彻（Fletcher）和鲍维尔加以发展和完善，成为现代公认的较好的算法之一。**

**DFP法是基于牛顿法的思想又作了重要改进。这种算法仅用到梯度，不必计算海赛阵及其逆矩阵，但又能使搜索方向逐渐逼近牛顿方向，具有较快的收敛速度。**

## 1. 基本思想

变量的尺度变换是放大或缩小各个坐标。通过尺度变换可以把函数的偏心程度降到最低限度。

例如在用最速下降法求  $f(x) = x_1^2 + 25x_2^2$  的极小

值时,需要进行10次迭代才能达到极小点  $x = [0,0]^T$

如作变换  $y_1 = x_1, \quad y_2 = 5x_2$

把  $x_2$  的尺度放大5倍, 则目标函数等值线由一簇椭圆变成一簇同心圆。

$$\phi(y_1, y_2) = y_1^2 + y_2^2$$

消除了函数的偏心，用最速下降法只需一次迭代即可求得极小点。

梯度法构造简单，只用到一阶偏导数，计算量小，初始点可任选，且开始几次迭代，目标函数值下降很快；其主要缺点是迭代点接近 $X^*$ 时，即使对二次正定函数收敛也非常慢。

牛顿法收敛很快，对于二次函数只需迭代一次便达到最优点，对非二次函数也能较快迭代到最优点，但要计算二阶偏导数矩阵及其逆阵，对维数较高的优化问题，其计算工作和存储量都太大。

*能不能将两种算法的优点综合起来，扬长避短？*

利用牛顿法的迭代公式，然而并不计算 $H^{-1}(\underline{x}^k)$ ，而是用一个对称，正定矩阵 $A^k$ 近似代替 $H^{-1}(\underline{x}^k)$

牛顿法中： $\underline{d}^k = -H^{-1}(\underline{x}^k) \nabla f(\underline{x}^k)$

变尺度法中：构造一个对称，正定的矩阵 $A^k$ 来代替 $H^{-1}(\underline{x})$

$$\underline{d}^k = -A^k \nabla f(\underline{x}^k)$$

$A^{k+1} = A^k + \Delta A^k$ （称为修正矩阵）

当 $k=0$ ， $A^k = I$ ，此时 $\underline{d}^k = -\nabla f(\underline{x}^k)$ （负梯度方向）

当 $\underline{x}^k \rightarrow \underline{x}^*$ 时， $A^k \rightarrow H^{-1}(\underline{x}^k)$

则 $\underline{d}^k = -H^{-1}(\underline{x}^k) \nabla f(\underline{x}^k)$ （牛顿方向）

在迭代开始时为负梯度方向，后来为牛顿方向

$A^k$ 在不断变化，故称为变尺度法



## 2. 构造尺度矩阵 $A^k$

从初始矩阵 $A^0=I$ (单位矩阵)开始, 通过对公式

$$A^{k+1} = A^k + \Delta A^k$$

中修正矩阵 $\Delta A^k$ 的不断修正, 在迭代中逐步逼近于 $G^{-1}(x^k)$ 。

因此, 一旦达到最优点附近, 就可望达到牛顿法的收敛速度。

### 1) DFP法 (Davidon-Fletcher-Powell)

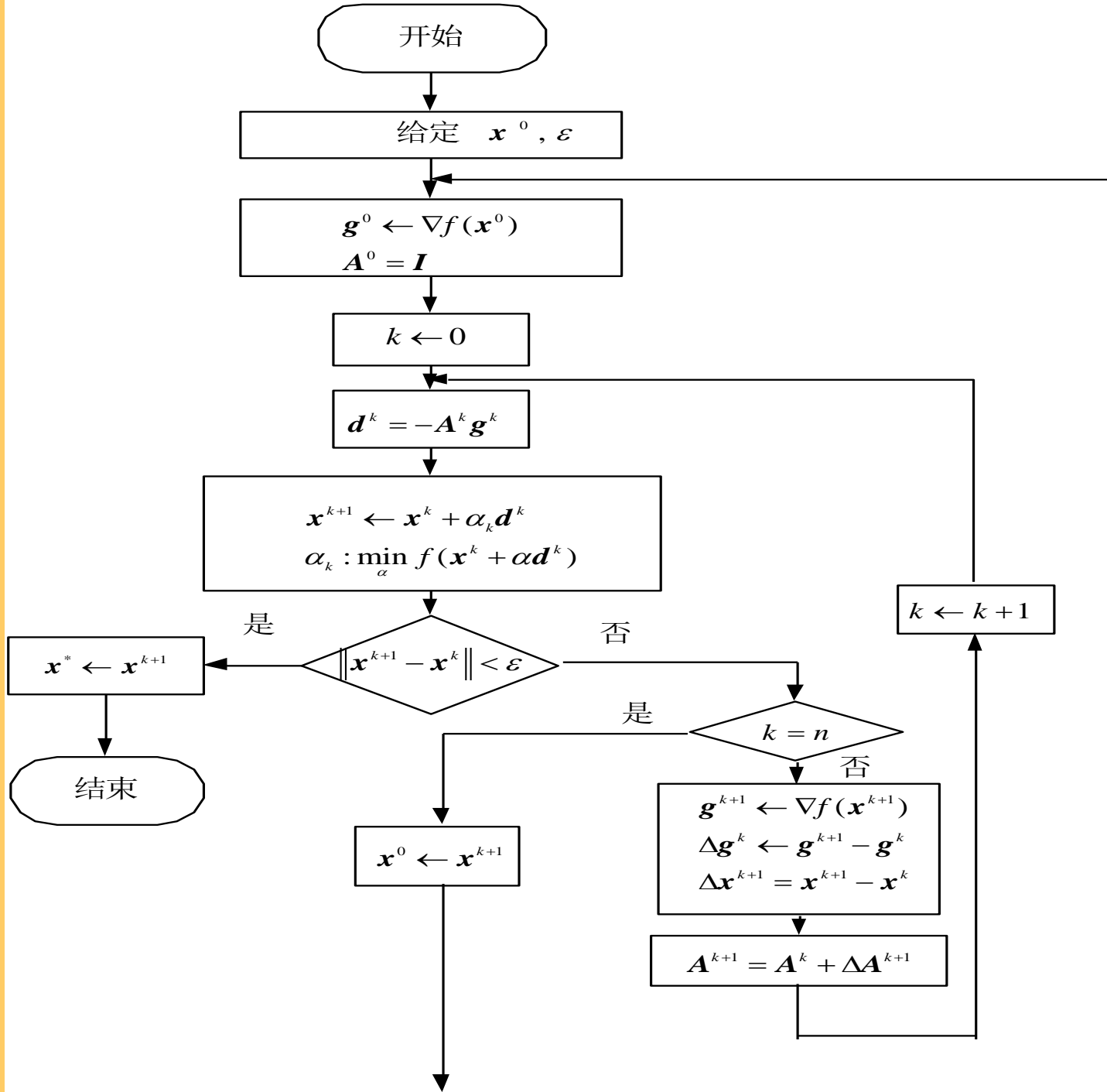
$$\Delta A^k = \frac{\Delta x^k [\Delta x^k]^T}{[\Delta x^k]^T \Delta g^k} - \frac{A^k \Delta g^k [\Delta g^k]^T A^k}{[\Delta g^k]^T A^k \Delta g^k}$$

$$\text{式中} \quad \begin{cases} \Delta g^k = g^{k+1} - g^k = \nabla f(x^{k+1}) - \nabla f(x^k) \\ \Delta x^k = x^{k+1} - x^k \end{cases}$$

## 2) BFGS算法(Broyden-Fletcher-Goldfrob-Shanno)

$$\Delta A^k = \frac{1}{[\Delta \mathbf{x}^k]^T \Delta \mathbf{g}^k} \left\{ \Delta \mathbf{x}^k [\Delta \mathbf{x}^k]^T + \frac{\Delta \mathbf{x}^k [\Delta \mathbf{x}^k]^T [\Delta \mathbf{g}^k]^T A^k \Delta \mathbf{g}^k}{[\Delta \mathbf{x}^k]^T \Delta \mathbf{g}^k} - A^k \Delta \mathbf{g}^k [\Delta \mathbf{x}]^T - \Delta \mathbf{x}^k [\Delta \mathbf{g}^k]^T A^k \right\}$$

**DFP**算法由于舍入误差和一维搜索不精确，有可能导致构造矩阵的正定性遭到破坏，以至算法不稳定。**BFGS**算法对于维数较高问题具有更好的稳定性。



例： 用**DFP**算法求下列问题的极值：

$$f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$$

- 解： 1) 取初始点 $\mathbf{x}^0 = [1 \ 1]^T$ ，为了按**DFP**法构造第一次搜寻方向 $\mathbf{d}^0$ ，需计算初始点处的梯度

$$\mathbf{g}^0 = \nabla f(\mathbf{x}^0) = \begin{bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{bmatrix}_{\mathbf{x}^0} = \begin{bmatrix} -4 \\ 2 \end{bmatrix}$$

取初始变尺度矩阵为单位矩阵 $\mathbf{A}^0 = \mathbf{I}$ ，则第一次搜寻方向为

$$\mathbf{d}^0 = -\mathbf{A}^0 \mathbf{g}^0 = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -4 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$$

沿 $d^0$ 方向进行一维搜索，得

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha_0 \mathbf{d}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha_0 \begin{bmatrix} 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 + 4\alpha_0 \\ 1 - 2\alpha_0 \end{bmatrix}$$

$\alpha_0$  为一维搜索最佳步长，应满足

$$f(\mathbf{x}^1) = \min_{\alpha} f(\mathbf{x}^0 + \alpha \mathbf{d}^0) = \min_{\alpha} (40\alpha^2 - 20\alpha - 3)$$

得：

$$\alpha_0 = 0.25, \quad \mathbf{x}^1 = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix}$$

2) 再按DFP法构造点 $\mathbf{x}^1$ 处的搜寻方向 $\mathbf{d}^1$ ，需计算

$$\mathbf{g}^1 = \begin{bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{bmatrix}_{\mathbf{x}^1} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

$$\Delta \mathbf{g}^0 = \mathbf{g}^1 - \mathbf{g}^0 = \begin{bmatrix} -1 \\ -2 \end{bmatrix} - \begin{bmatrix} -4 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$

$$\Delta \mathbf{x}^0 = \mathbf{x}^1 - \mathbf{x}^0 = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}$$

代入校正公式

$$\Delta \mathbf{A}^0 = \frac{\Delta \mathbf{x}^0 [\Delta \mathbf{x}^0]^T}{[\Delta \mathbf{x}^0]^T \Delta \mathbf{g}^0} - \frac{\mathbf{A}^0 \Delta \mathbf{g}^0 [\Delta \mathbf{g}^0]^T \mathbf{A}^0}{[\Delta \mathbf{g}^0]^T \mathbf{A}^0 \Delta \mathbf{g}^0}$$

$$= \frac{\begin{bmatrix} 1 \\ -0.5 \end{bmatrix} \begin{bmatrix} 1 & -0.5 \end{bmatrix}}{\begin{bmatrix} 1 & -0.5 \end{bmatrix} \begin{bmatrix} 3 \\ -4 \end{bmatrix}} - \frac{\begin{bmatrix} 3 \\ -4 \end{bmatrix} \begin{bmatrix} 3 & -4 \end{bmatrix}}{\begin{bmatrix} 3 & -4 \end{bmatrix} \begin{bmatrix} 3 \\ -4 \end{bmatrix}}$$

$$\begin{aligned}
 \mathbf{A}^1 &= \mathbf{A}^0 + \Delta \mathbf{A}^0 \\
 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{5} \begin{bmatrix} 1 & -0.5 \\ -0.5 & 0.25 \end{bmatrix} - \frac{1}{25} \begin{bmatrix} 9 & -12 \\ -12 & 16 \end{bmatrix} = \begin{bmatrix} \frac{21}{25} & \frac{19}{50} \\ \frac{19}{50} & \frac{41}{100} \end{bmatrix}
 \end{aligned}$$

第二次搜寻方向为

$$\mathbf{d}^1 = -\mathbf{A}^1 \mathbf{g}^1 = \begin{bmatrix} 8 \\ 6 \\ 6 \\ 5 \end{bmatrix}$$

再沿 $\mathbf{d}^1$ 进行一维搜索，得

$$\mathbf{x}^2 = \mathbf{x}^1 + \alpha_1 \mathbf{d}^1 = \begin{bmatrix} 2 + \frac{8}{5} \alpha_1 \\ 0.5 + \frac{6}{5} \alpha_1 \end{bmatrix}$$

$\alpha_1$  为一维搜索最佳步长，应满足

$$f(\mathbf{x}^2) = \min_{\alpha} f(\mathbf{x}^1 + \alpha \mathbf{d}^1) = \min_{\alpha} \left( \frac{8}{5} \alpha^2 - 4\alpha - \frac{11}{2} \right)$$

得  $\alpha_1 = \frac{5}{4}$  ,  $\mathbf{x}^2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$

3) 判断 $\mathbf{x}^2$ 是否为极值点

梯度:

$$\nabla f(\mathbf{x}^2) = \begin{bmatrix} 2x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 \end{bmatrix}_{\mathbf{x}^2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

海赛矩阵:

$$\nabla^2 f(\mathbf{x}^2) = \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix}$$



梯度为零向量, 海赛矩阵正定。可见点满足极值充要条件, 因此为极小点。

$$\mathbf{x}^* = \mathbf{x}^2 = \begin{bmatrix} 4 & 2 \end{bmatrix}^T$$

$$f(\mathbf{x}^*) = -8$$

### 3 变尺度法的特点

#### (1) 收敛速度快

开始为负梯度方向（下降快）

接近最优点时为牛顿方向（收敛快）

#### (2) 计算量小，不计算 $-H^{-1}(\underline{x}^K)$

#### (3) 算法稳定性好，特别是BFGS法

# 无约束优化方法 ——间接法总结

## 1、梯度法

方向 负梯度 用到一阶导数

适合于精度不高或用于复杂函数寻找一个好的初始点

## 2、牛顿法

用到一阶导数和海色矩阵，具有二次收敛性

要求海色矩阵奇异，且维数不宜太高

## 3、变尺度法

收敛快，效果好，被认为是目前最有效的无约束优化方法。适用于维数较高，具有一阶偏导数的目标函数

## 作业：

1。用梯度法求解 $\min f(\underline{X}) = (x_1 - 5)^2 + (x_2 - 4)^2$ 的极小点，

$$\underline{X}^{(0)} = \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

2。变尺度法的特点是什么？

## 4-5 无约束极值问题的直接数值解法

## 1 直接数值解法的特点

- (1) 不要求目标函数有较好的解析性质
- (2) 求解优化问题的成功率（即可靠性）比解析法高
- (3) 由于没有利用函数的解析性质，导致收敛速度较慢，计算时间长

## 2 常用直接数值解法有以下几种

坐标轮换法

模式搜索法（基本共轭法）

Powell 共轭法

单形替换法

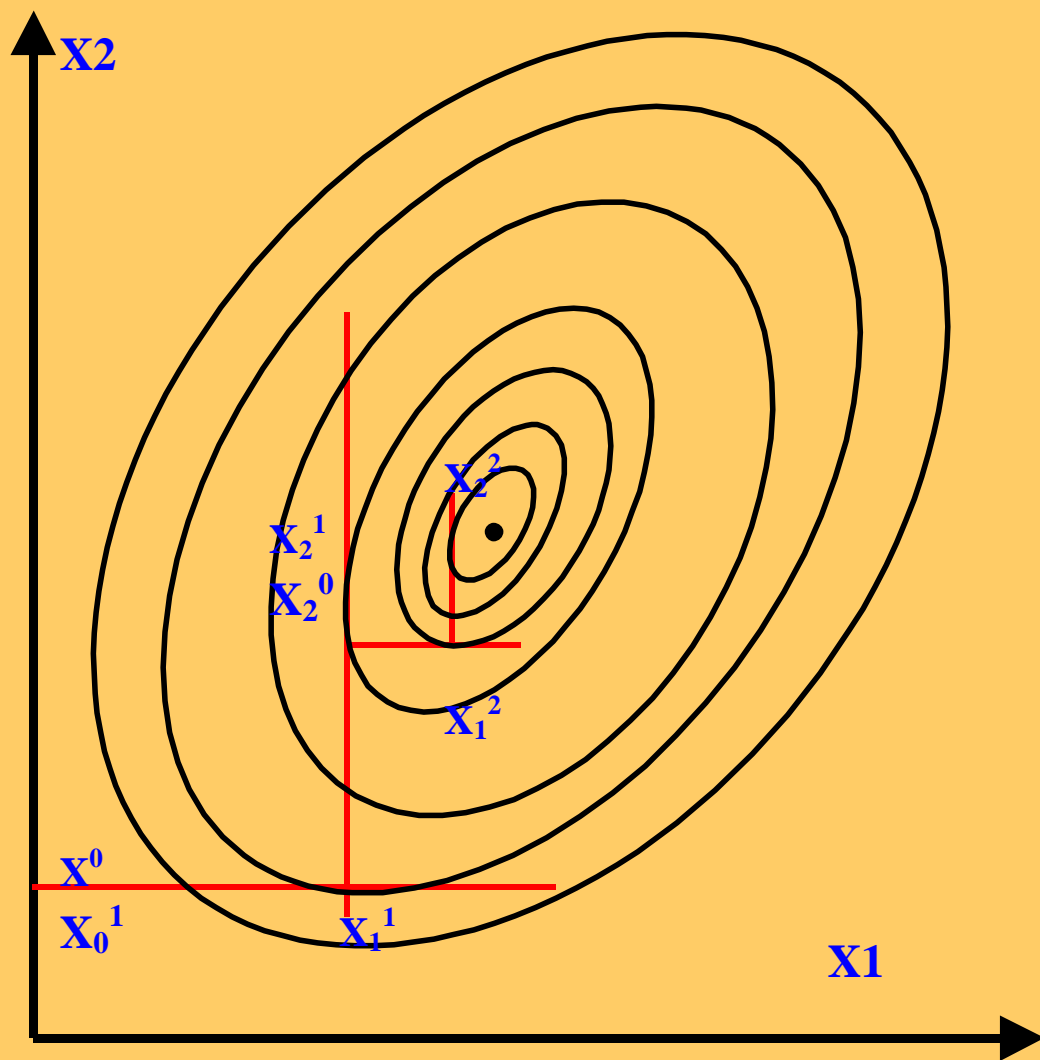
## 4-6 坐标轮换法（变量轮换法）

# 1 基本原理

将一个多维无约束优化问题转化为一系列一维优化问题来求解，即依次沿着坐标轴的方向进行一维搜索，求得最小点



## 2 步骤



(1) 从  $x_0^1$  出发, 沿坐标轴  $x_1$  方向一维搜索, 即  $\underline{d}_1^1 = [1, 0]^T$ , 得到  $x_1^1$

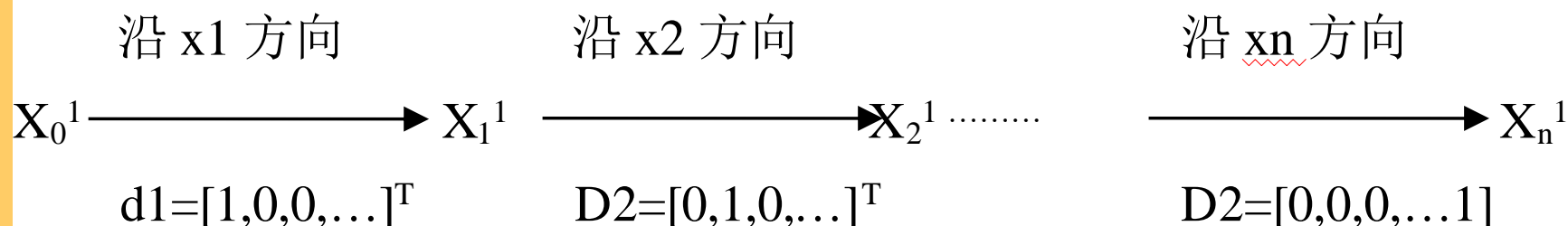
(2) 从  $x_1^1$  出发, 沿坐标轴  $x_2$  方向一维搜索, 即  $\underline{d}_2^1 = [0, 1]^T$ , 得到  $x_2^1$

(3) 若第一轮计算还未收敛, 则开始下一轮计算,  $x_0^2 \Leftarrow x_2^1$

继续下去

(沿坐标轴方向轮换进行一维搜索)

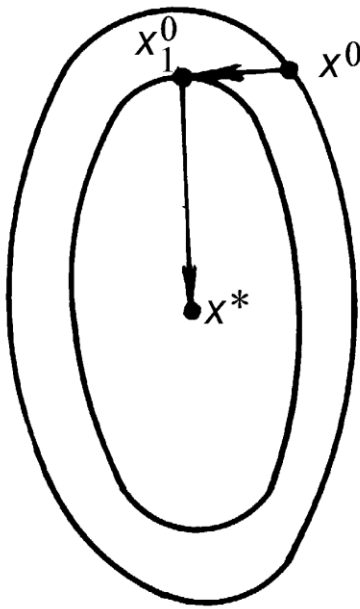
## 推广到n维



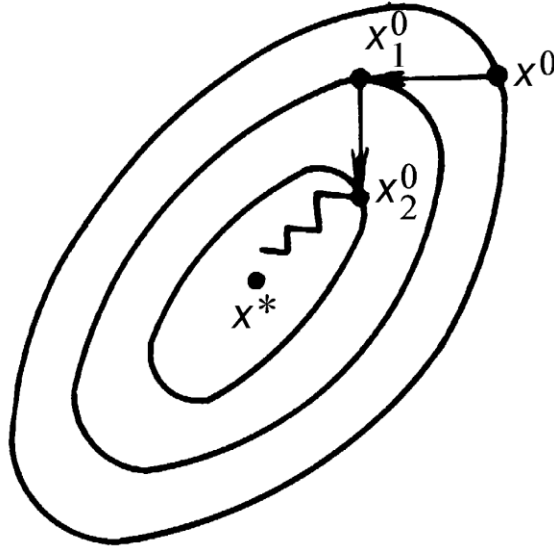
若不收敛，则令  $X_0^2 \leftarrow X_n^1$ ，再进行第二次的循环

### 3 特点

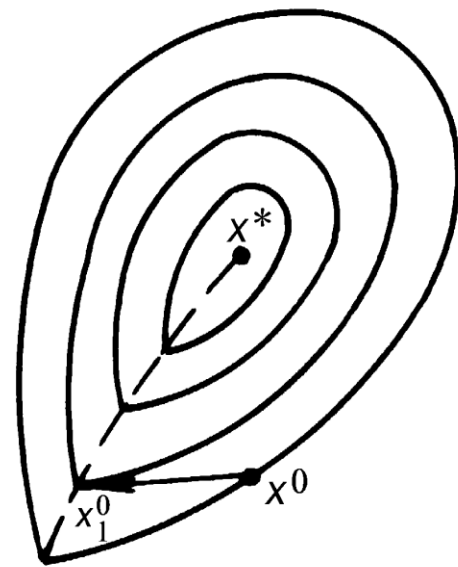
- (1) 将n维问题转化为了一维问题
- (2) 效率低：因搜索方向始终平行于坐标轴，因此不能适应函数变化的情况
- (3) 收敛快慢和函数形状有很大的关系



a)



b)



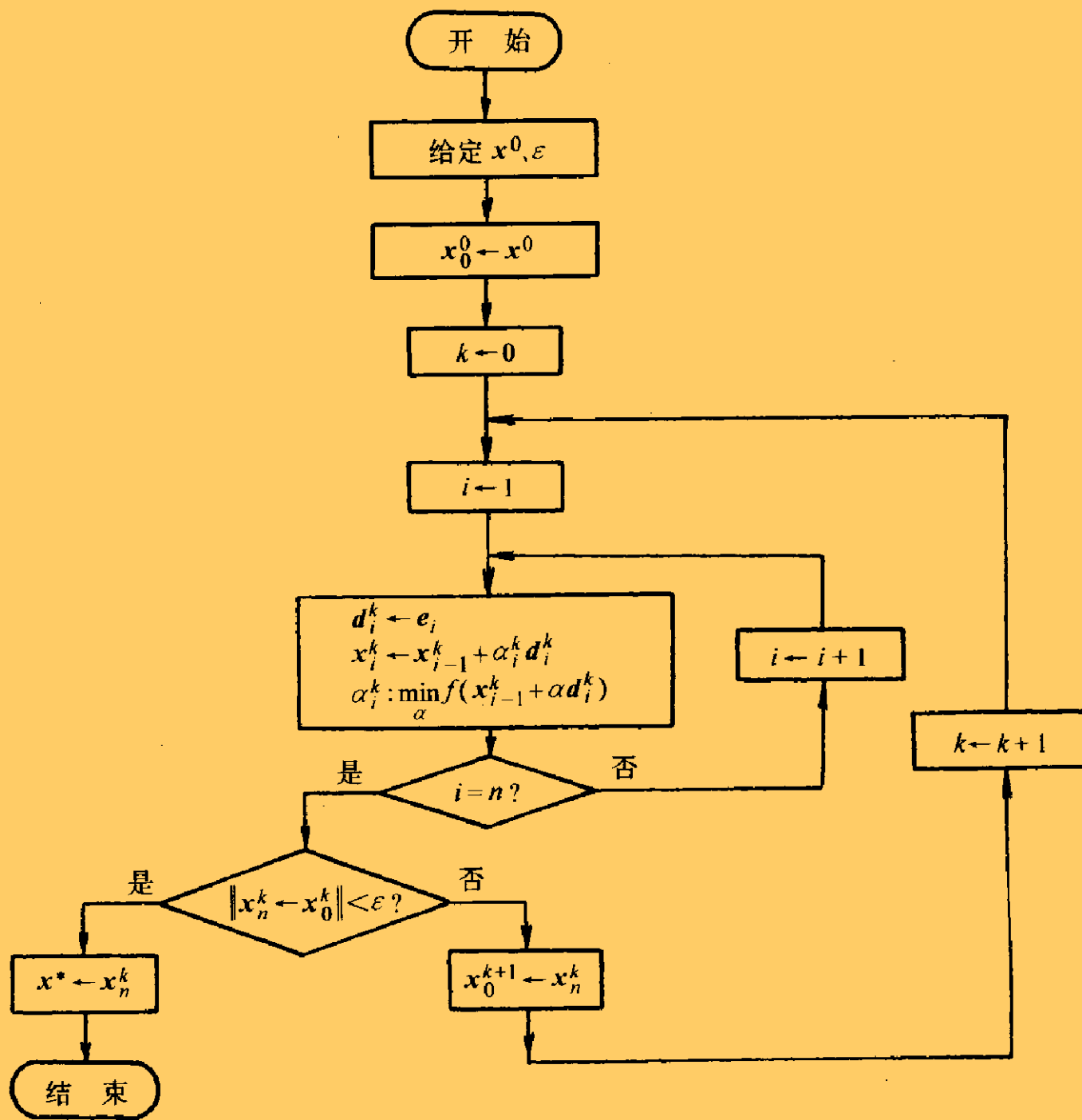
c)

(4) 计算量少，程序简单，不需要求函数导数的直接探索目标函数最优解的方法；

(5) 探索路线较长，问题的维数愈多求解的效率愈低。当维数 $n > 10$ 时，则不应采用此法。仅适用于 $n$ 较少（ $n < 10$ ）的目标函数求优；

(6) 改变初始点重新迭代，可避免出现病态。

### 3 程序框图

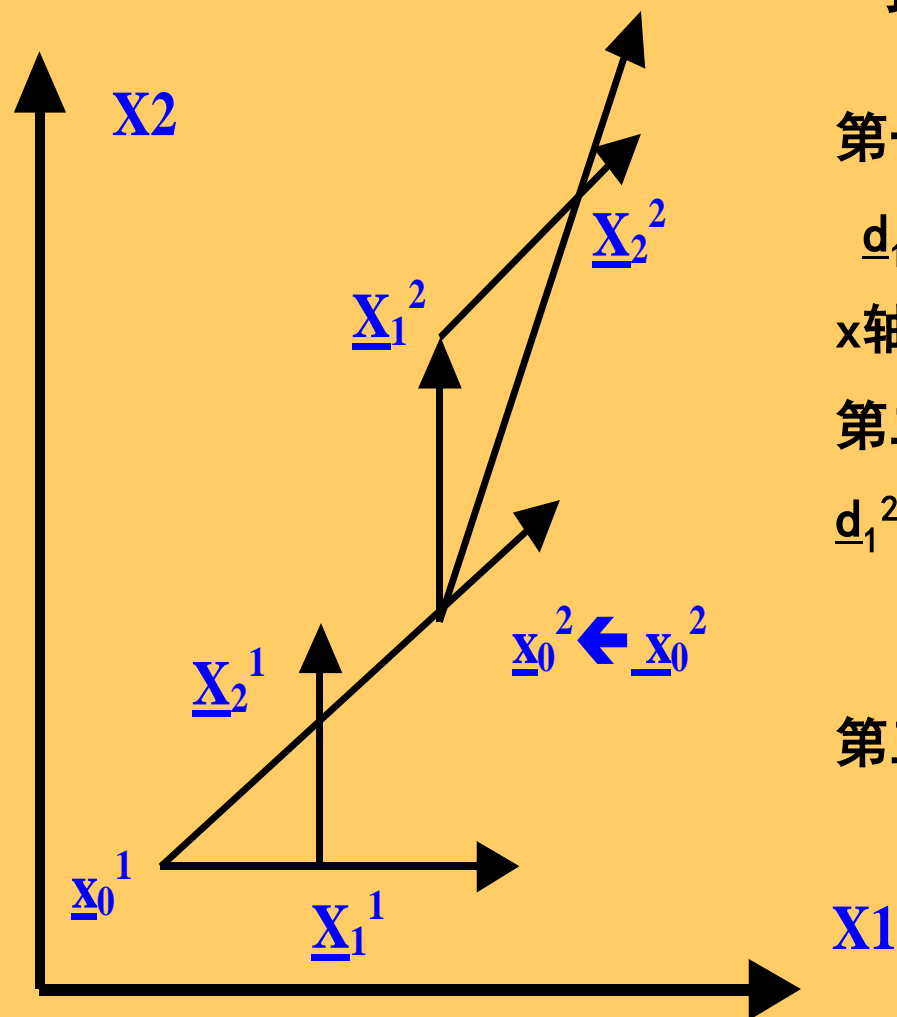


## 4-7 模式搜索法（基本共轭方向法）

## 1. 基本原理

首先采用坐标轮换法进行第一轮迭代，然后以第一轮迭代的最末一个点和初始点相连，构成一个新的方向，并以此新方向为最末一个方向，而去掉第一个方向，得到第二轮的迭代方向

## 2 步骤(以二维为例)



## 搜索方向

第一轮:

$\underline{d}_1^1$

x轴

$\underline{d}_2^1$

y轴

$\underline{d}^1$

第一轮模式搜索方向

第二轮:

$\underline{d}_1^2 \leftarrow \underline{d}_2^1$     $\underline{d}_2^2 \leftarrow \underline{d}^1$     $\underline{d}^2$

第二轮模式搜索方向

第三轮:  $\underline{d}_1^3 \leftarrow \underline{d}^1$     $\underline{d}_2^3 \leftarrow \underline{d}^2$



**把二维情况的算法扩展到n维，要点是：**

**在每一轮迭代中总有一个始点（第一轮的始点是任选的初始点）和n个线性独立的搜索方向。从始点出发顺次沿n个方向做一维搜索得一终点，由始点和终点决定了一个新的搜索方向。由始点和终点决定一个新的搜索方向。用这个方向替换原来n个方向中的一个，于是形成新的搜索方向**

**替换的原则是去掉原方向组的一个方向而将新方向排在原方向的最后。**

**从这一轮的搜索终点出发沿新的搜索方向作一维搜索而得到的极小点，作为下一轮的始点**

### ●3.共轭方向:

● 设 $G$ 为 $n \times n$ 阶实对称正定矩阵, 如果有两个 $n$ 维向量 $d^0$ 和 $d^1$ 满足  $(d^0)^T G d^1 = 0$ , 则称向量 $d^0$ 与 $d^1$ 关于矩阵 $G$ 共轭。

当 $G$ 为单位矩阵时,  $(d^0)^T d^1 = 0$

假设目标函数 $f(x)$  在极值点附近的二次近似函数为

$$f(x) = \frac{1}{2} x^T G x + b^T x + c$$

对二维情况

任选取初始点 $x^0$ 沿某个下降方向 $d^0$ 作一维搜索, 得 $x^1$

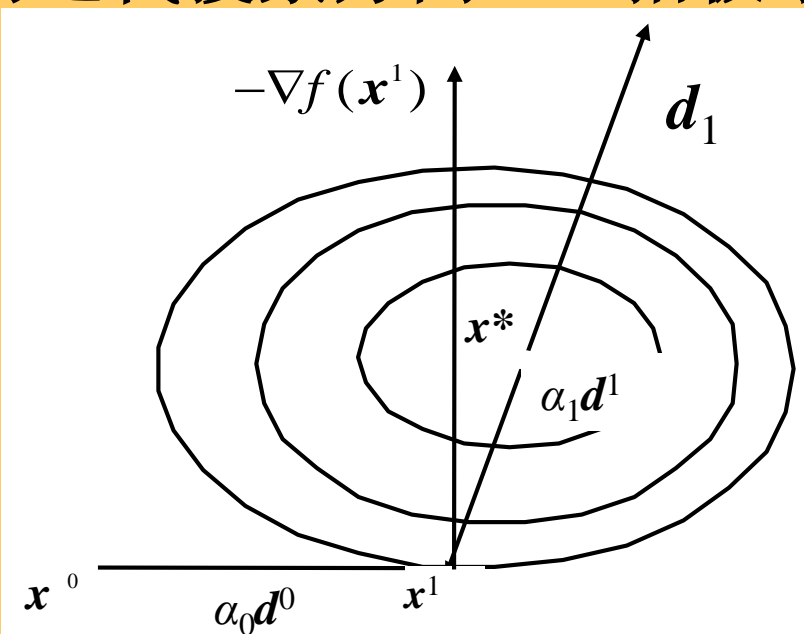
$$x^1 = x^0 + \alpha_0 d^0$$

因为  $\alpha_0$  是沿  $d^0$  方向搜索的最佳步长，即在点  $x^1$  处函数  $f(x)$  沿方向  $d^0$  的方向导数为零。考虑到点  $x^1$  处方向导数与梯度之间的关系，故有

$$\left. \frac{\partial f}{\partial d^0} \right|_{x^1} = [\nabla f(x^1)]^T d^0 = 0$$

如果按最速下降法，选择负梯度方向  $-\nabla f(x^1)$  为搜索方向，则将发生锯齿现象。

取下一一次的迭代搜索方向  $d^1$  直指极小点  $x^*$ 。



如果能够选定这样的搜索方向，那么对于二元二次函数只需顺次进行 $d^0$ 、 $d^1$ 两次直线搜索就可以求到极小点 $x^*$ ，即有

$$x^* = x^1 + \alpha_1 d^1$$

那么这样的 $d^1$ 方向应该满足什么条件呢？

对于前述的二次函数： $f(x) = \frac{1}{2} x^T G x + b^T x + c$

有  $\nabla f(x^1) = Gx^1 + b$

当  $x^1 \neq x^*$  时，  $\alpha_1 \neq 0$

$x^*$ 是 $f(x)$ 极小点，应满足极值必要条件，故有

$$\nabla f(x^*) = Gx^* + b = 0$$

$$\nabla f(x^*) = G(x^1 + \alpha_1 d^1) + b = \nabla f(x^1) + \alpha_1 Gd^1 = 0$$

将等式两边同时左乘  $(d^0)^T$  得： $(d^0)^T Gd^1 = 0$

$$(d^0)^T G d^1 = 0$$

$$(d^0)^T \nabla^2 f(x) d^1 = 0$$

就是使 $d^1$ 直指极小点 $x^*$ ， $d^1$ 所必须满足的条件。

两个向量 $d^0$ 和 $d^1$ 称为 $G$ 的共轭向量，或称 $d^0$ 和 $d^1$ 对 $G$ 是共轭方向。

## 2. 共轭方向的性质

性质1 若非零向量系 $d^0, d^1, d^2, \dots, d^{m-1}$ 是对 $G$ 共轭，则这 $m$ 个向量是线性无关的。

性质2 在 $n$ 维空间中互相共轭的非零向量的个数不超过 $n$ 。

性质3 从任意初始点出发，顺次沿 $n$ 个 $G$ 的共轭方向 $d^0, d^1, d^2, \dots$ ，进行一维搜索，最多经过 $n$ 次迭代就可以找到的二次函数 $f(x)$ 极小点。

## 4-8 Powell 共轭方向法

鲍威尔法是以共轭方向为基础的收敛较快的直接法之一，是一种十分有效的算法。

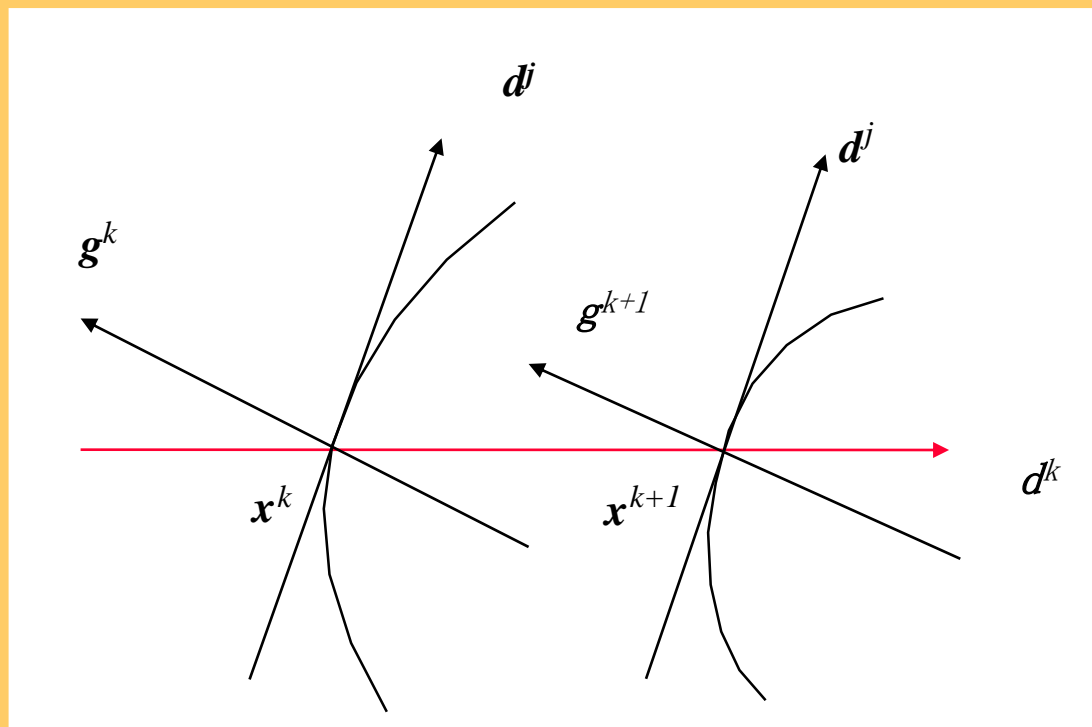
1964年，鲍威尔提出这种算法，其基本思想是直接利用迭代点的目标函数值来构造共轭方向，然后从任一初始点开始，逐次沿共轭方向作一维搜索求极小点。并在以后的实践中进行了改进。

对函数：
$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{b}^T \mathbf{x} + \mathbf{c}$$

基本思想：在不用导数的前提下，在迭代中逐次构造  $G$  的共轭方向。

## 1. 共轭方向的生成

设  $\mathbf{x}^k$ ,  $\mathbf{x}^{k+1}$  为从不同点出发，沿同一方向  $\mathbf{d}^j$  进行一维搜索而到的两个极小点。





梯度和等值面相垂直的性质,  $\mathbf{d}^j$ 和  $\mathbf{x}^k$ ,  $\mathbf{x}^{k+1}$ 两点处的梯度 $\mathbf{g}^k$ ,  $\mathbf{g}^{k+1}$ 之间存在关系:

$$(\mathbf{d}^j)^T \mathbf{g}^k = 0, \quad (\mathbf{d}^j)^T \mathbf{g}^{k+1} = 0$$

另一方面, 对于上述二次函数, 其 $\mathbf{x}^k$ ,  $\mathbf{x}^{k+1}$ 两点处的梯度可表示为:

$$\mathbf{g}^k = \mathbf{G}\mathbf{x}^k + \mathbf{b}, \quad \mathbf{g}^{k+1} = \mathbf{G}\mathbf{x}^{k+1} + \mathbf{b}$$

因而有  $(\mathbf{d}^j)^T (\mathbf{g}^k - \mathbf{g}^{k+1}) = (\mathbf{d}^j)^T \mathbf{G}(\mathbf{x}^{k+1} - \mathbf{x}^k) = 0$

取  $\mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$

这说明只要沿 $\mathbf{d}^j$ 方向分别对函作两次一维搜索, 得到两个极小点 $\mathbf{x}^k$ 和 $\mathbf{x}^{k+1}$ , 那么这两点的连线所给出的方向 $\mathbf{d}^k$ 就是与 $\mathbf{d}^j$ 一起对 $\mathbf{G}$ 共轭的方向。

## 2. 关键问题

因为在迭代中的 $n$ 个搜索方向有时会变成线性相关而不能形成共轭方向。这时组不成 $n$ 维空间，可能求不到极小点，所以上述基本算法有待改进。

在基本共轭方向算法中，每一轮迭代都用连结始点和终点所产生出的搜索方向去替换原向量组中的第一个向量，而不管它的“好坏”，这是产生向量组线性相关的原因所在。

在改进的算法中首先判断原向量组是否需要替换。如果需要替换，还要进一步判断原向量组中哪个向量最坏，然后再用新产生的向量替换这个最坏的向量，以保证逐次生成共轭方向。

## 关键问题

在Powell方法中，每一轮迭代都用连结始点和终点所产生的搜索方向去替换原方向组的第一个向量，而不管它的“好坏”，这是产生向量组线性相关的原因所在，因此在改进的算法中首先判断原向量组是否需要替换

若第K次迭代时，方向组是 $[d_1^K, d_2^K, \dots, d_n^K]$ 以及产生的模式方向 $d^K$ ，那么：

- 1) 在第k+1次迭代时，原方向组是否需要更新？若要，则
- 2) 原方向组中的哪个方向 $d_m^K$  (即 $m=?$ ) 需被替换？保证新的方向“最接近共轭”

为此，要解决两个关键问题：

- (1)  $d^{k+1}$ 是否较好？是否应该进入新的方向组？  
即方向组是否进行更新？
- (2) 如果应该更新方向组， $d^{k+1}$ 不一定替换方向  $d_1^k$ ，而是有选择地替换某方向  $d_m^k$ 。

令在 $k$ 次循环中  $F_0 = f(\mathbf{x}_0^k)$

$$F_2 = f(\mathbf{x}_n^k)$$

$$F_3 = f(\mathbf{x}_{n+1}^k)$$

$$\mathbf{x}_0^k, \mathbf{x}_n^k, \mathbf{x}_{n+1}^k \quad ( \mathbf{x}_{n+1}^k = \mathbf{x}_n^k + (\mathbf{x}_n^k - \mathbf{x}_0^k) = 2\mathbf{x}_n^k - \mathbf{x}_0^k )$$

分别称为一轮迭代的始点、终点和反射点。

记:  $f_i = f(\mathbf{x}_i^k)$  ( $i = 0, 1, 2, \dots, n$ ) 因此  $F_0 = f_0$ ,  $F_2 = f_n$

$$\Delta_i = f_{i-1} - f_i \quad (i = 1, 2, \dots, n)$$

则在循环中函数下降最多的第 $m$ 次迭代是

$$\Delta_m = \max_{1 \leq i \leq n} \Delta_i = f_{m-1} - f_m$$

相应的方向为 $d_m^k$ 。

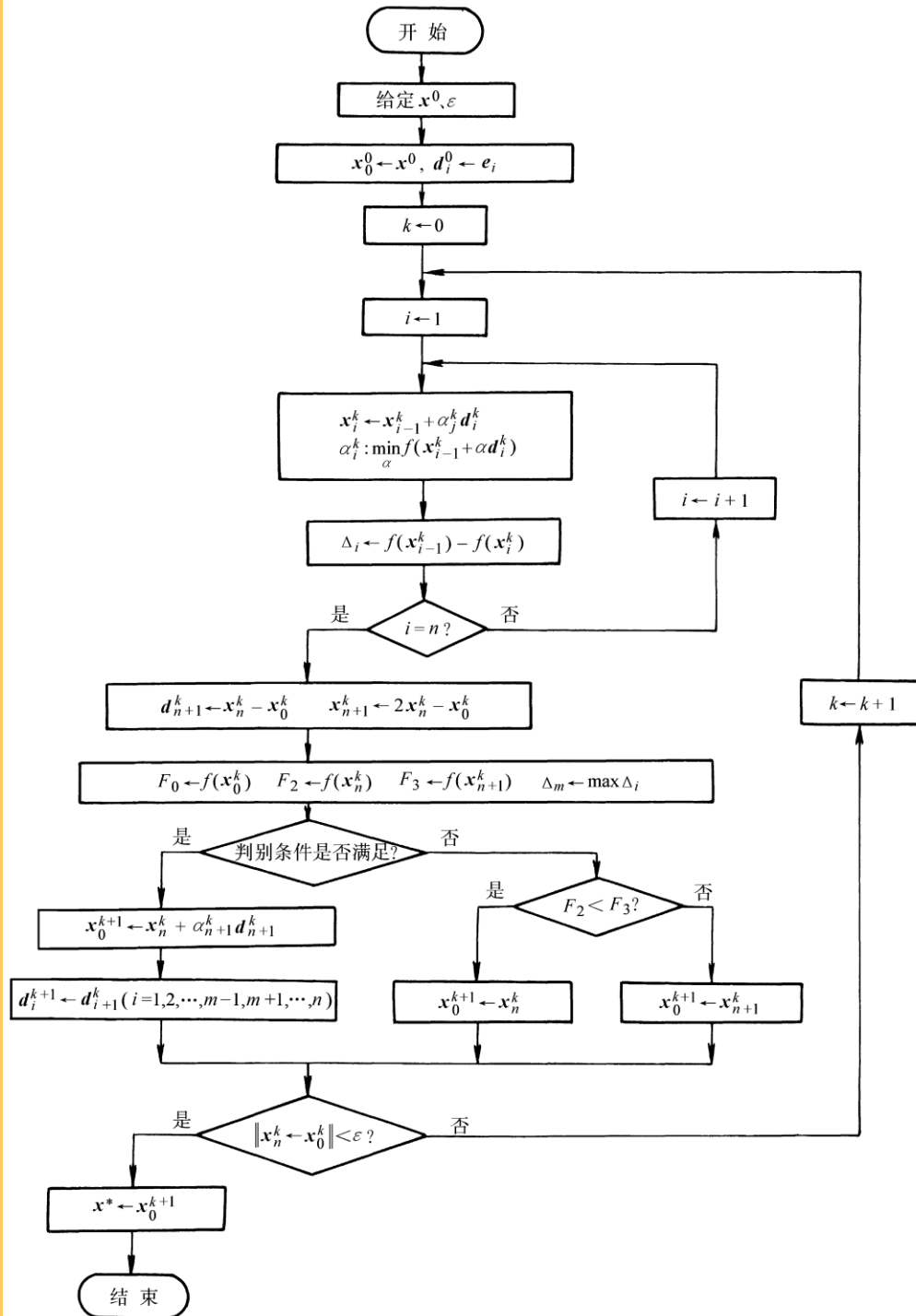
为了构成共轭性好的方向组，须遵循下列准则：在 $k$ 次循环中，若满足条件：

$$F_3 < F_0$$

$$\text{和 } (F_0 - 2F_2 + F_3)(F_0 - F_2 - \Delta_m)^2 < 0.5\Delta_m(F_0 - F_3)^2$$

则选用新方向 $d^k$ ，并在第 $k+1$ 迭代中用 $d^k$ 替换对应于 $\Delta_m$ 的方向  $d_m^k$ 。否则，仍然用原方向组进行第 $k+1$ 迭代。

这样重复迭代的结果，后面加进去的向量都彼此对 $G$ 共轭，经 $n$ 轮迭代即可得到一个由 $n$ 个共轭方向所组成的方向组。对于二次函数，最多 $n$ 次就可找到极小点，而对一般函数，往往要超过 $n$ 次才能找到极小点（这里“ $n$ ”表示设计空间的维数）。



### 例4-5 用改进的鲍威尔法求目标函数

$$f(\mathbf{x}) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$$

的最优解。已知初始点  $[1, 1]^T$ ，迭代精度

$$\varepsilon = 0.001。$$

解：（1）第1轮迭代计算

$$\mathbf{x}_0^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad F_0 = f_0 = f(\mathbf{x}_0^0) = -3$$

沿  $\mathbf{e}_1$  方向进行一维搜索

$$\min f(\mathbf{x}_0^0 + \alpha \mathbf{e}_1) = \alpha^2 - 4\alpha - 3$$

$$\text{得} \quad \alpha_1 = 2$$

$$\mathbf{x}_1^0 = \mathbf{x}_0^0 + \alpha_1 \mathbf{e}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad f_1 = f(\mathbf{x}_1^0) = -7$$



以  $\mathbf{x}_1^0$  为起点，沿第二坐标轴方向  $\mathbf{e}_2$  进行一维搜索

$$\min f(\mathbf{x}_1^0 + \alpha \mathbf{e}_2) = 2\alpha^2 - 2\alpha - 7$$

得  $\alpha_2 = 0.5$

$$\mathbf{x}_2^0 = \mathbf{x}_1^0 + \alpha_2 \mathbf{e}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + 0.5 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1.5 \end{bmatrix}$$

$$f_2 = f(\mathbf{x}_2^0) = -7.5$$

确定此轮中的最大下降量及其相应方向

$$\Delta_1 = f_0 - f_1 = f(\mathbf{x}_0^0) - f(\mathbf{x}_1^0) = 4$$

$$\Delta_2 = f_1 - f_2 = f(\mathbf{x}_1^0) - f(\mathbf{x}_2^0) = 0.5$$

$$\Delta_m = \max[\Delta_1, \Delta_2] = 4$$

反射点及其函数值

$$\mathbf{x}_3^0 = 2\mathbf{x}_2^0 - \mathbf{x}_0^0 = 2\begin{bmatrix} 3 \\ 1.5 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \quad F_3 = f(\mathbf{x}_3^0) = -7$$

检验Powell条件

$$F_3 = -7 < F_0 = -3$$

$$(F_0 - 2F_2 + F_3)(F_0 - F_2 - \Delta_m)^2 = 1.25 < 0.5\Delta_m(F_0 - F_3)^2 = 32$$

由于满足Powell条件，则淘汰函数值下降量最大的方向 $e_1$ ，下一轮的基本方向组为 $e_2, d_3^0$ 。

构成新的方向

$$d_3^0 = x_2^0 - x_0^0 = \begin{bmatrix} 3 \\ 1.5 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix}$$

沿  $d_3^0$  方向一维搜索得极小点和极小值

$$x^1 = \begin{bmatrix} 3.8 \\ 1.7 \end{bmatrix}, \quad f(x^1) = -7.9$$

此点为下轮迭代初始点。

按点距准则检验终止条件

$$\|x^1 - x_0^1\| = \sqrt{(3.8-1)^2 + (1.7-1)^2} = 2.886 > \varepsilon$$

需进行第二轮迭代机算。

## (2) 第2轮迭代计算

此轮基本方向组为 $e_2, d_3^0$ , 分别相当于 $d_1^1, d_2^1$ , 起始点为 $x_0^1 = x^1$ 。

沿 $e_2$ 方向进行一维搜索得

$$x_1^1 = \begin{bmatrix} 3.8 \\ 1.9 \end{bmatrix} \quad f_1 = f(x_1^1) = -7.98$$

以 $x_1^1$ 为起点沿 $d_3^0$ 方向一维搜索得

$$x_2^1 = \begin{bmatrix} 3.96 \\ 1.9 \end{bmatrix} \quad f_2 = f(x_2^1) = -7.996$$

确定此轮中函数值最大下降量及其相应方向

$$\Delta_1 = 0.08$$

$$\Delta_2 = 0.016$$

$$\Delta_m = \max[\Delta_1, \Delta_2] = 0.08$$

## 反射点及其函数值

$$\mathbf{x}_3^1 = 2\mathbf{x}_2^1 - \mathbf{x}_0^1 = 2 \begin{bmatrix} 3.96 \\ 1.94 \end{bmatrix} - \begin{bmatrix} 3.8 \\ 1.7 \end{bmatrix} = \begin{bmatrix} 4.12 \\ 2.18 \end{bmatrix}$$

$$F_3 = f(\mathbf{x}_3^1) = -7.964$$

检验Powell条件，淘汰函数值下降量最大的方向 $\mathbf{e}_2$ ，下一轮的基本方向组应为 $\mathbf{d}_3^0$ ， $\mathbf{d}_3^1$ 。

构成新的方向

$$\mathbf{d}_3^1 = \mathbf{x}_2^1 - \mathbf{x}_0^1 = \begin{bmatrix} 3.96 \\ 1.94 \end{bmatrix} - \begin{bmatrix} 3.8 \\ 1.7 \end{bmatrix} = \begin{bmatrix} 0.16 \\ 0.24 \end{bmatrix}$$

沿 $\mathbf{d}_3^1$ 方向进行一维搜索得

$$\mathbf{x}^2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$f(\mathbf{x}^2) = -8$$

## 检验终止条件

$$\|\mathbf{x}^2 - \mathbf{x}_0^2\| = \sqrt{(4 - 3.8)^2 + (2 - 1.7)^2} = 0.36 > \varepsilon$$

### (3) 第3轮迭代计算

此轮基本方向组为 $d_3^0, d_3^1$ ，起始点为 $\mathbf{x}_0^2 = \mathbf{x}^2$ ，先后沿 $d_3^0, d_3^1$ 方向，进行一维搜索，得

$$\mathbf{x}_1^2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \quad \mathbf{x}_2^2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

检验终止条件  $\|\mathbf{x}_2^2 - \mathbf{x}_0^2\| = 0 < \varepsilon$

故最优解  $\mathbf{x}^* = \begin{bmatrix} 4 \\ 2 \end{bmatrix} \quad f(\mathbf{x}^*) = -8$

实际上，前两轮迭代的  $d_3^1$  ,  $d_3^2$  为共轭方向，由于本例目标函数是二次函数，按共轭方向的二次收敛性，故前两轮的结果就是问题的最优解，但每一轮迭代都需要进行  $n+1$  次迭代。

## 4-9 单形替换法

基本思想:

单纯形替换法也是一种不使用导数的求解无约束极小化问题的直接搜索方法，与前面几种方法不同的是，单纯形替换法不是利用搜索方向从一个点迭代到另一个更优的点，而是从一个单纯形迭代到另一个更优的单纯形。



## 1. 基本原理

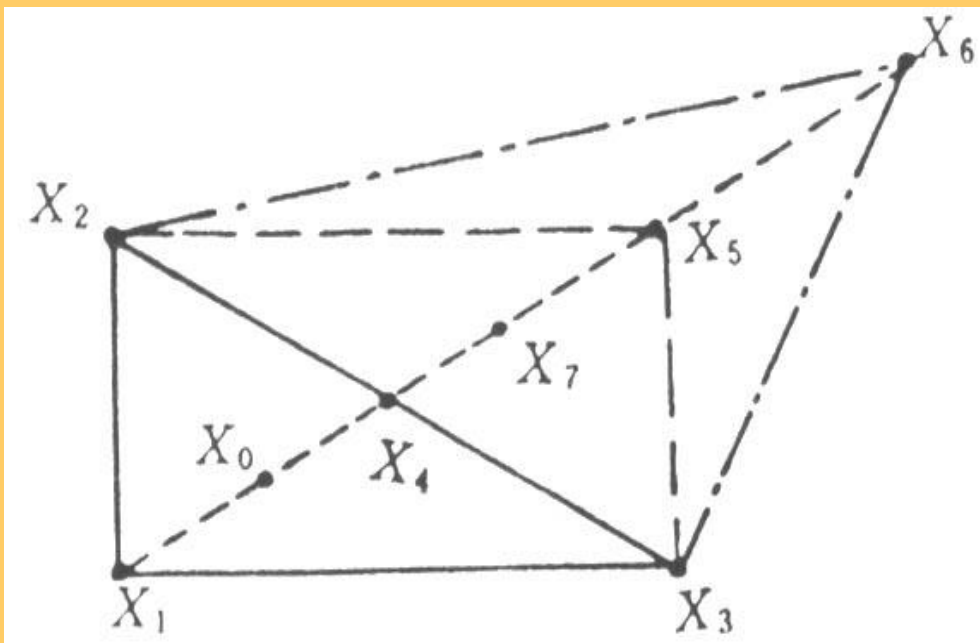
- 1) 计算单纯形各顶点的函数值
- 2) 确定搜索方向和步长
- 3) 找到一个好点替代原单纯形的坏点，构成新的单纯形

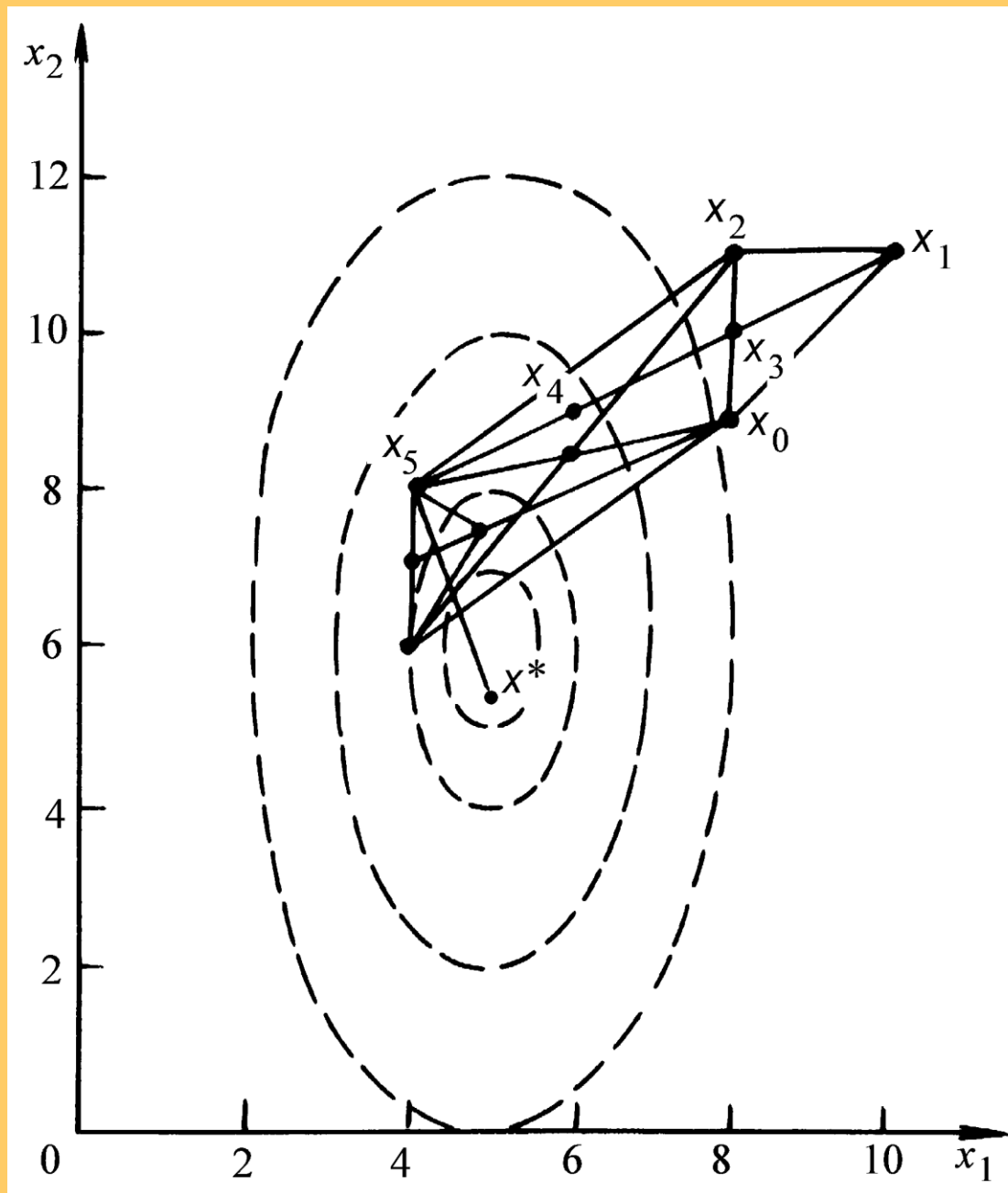
## 定义：单纯形

$n$ 维空间中的恰好有 $n+1$ 个顶点（极点）的有界的凸多面体称之为一个单纯形。

根据定义，可知，一维空间中的单纯形是线段，二维空间中的单纯形是三角形，而三维空间中的单纯形则是四面体。

在单纯形替换算法中，从一个单纯形到另一个单纯形的迭代主要通过反射、扩张、收缩和缩边这4个操作来实现。下面以二维问题为例来对4种操作进行说明（参见下图）。





## 2. 单形替换法特点:

a) 不必计算导数

b) 维数较高时, 迭代次数较多, 一般用于 $n < 10$ 的情况

# 无约束优化方法 ——直接法总结

## 1、坐标轮换法

计算效率较低

适合维数较低，目标函数无导数或导数较难求

## 2、Powell法

具有二次收敛性，收敛速度较快，可靠性高，被认为是直接法中最有效的方法之一

## 3、单纯形法

思路清楚，收敛慢

## 作业：

1. 坐标轮换法，模式搜索法（基本共轭方向法），Powell共轭方向法三者的异同点有哪些？
2. 单行替换法的基本原理是什么？

# 考试安排

**考试时间:**

**4月29日 第10周周五7:50-9点50**

**考试地点:**

**A104**