化工过程模拟及软件应用

杨鑫
化学化工学院
重庆理工大学
第一实验楼A203
cheyangxin@cqut.edu.cn

化工过程数值计算

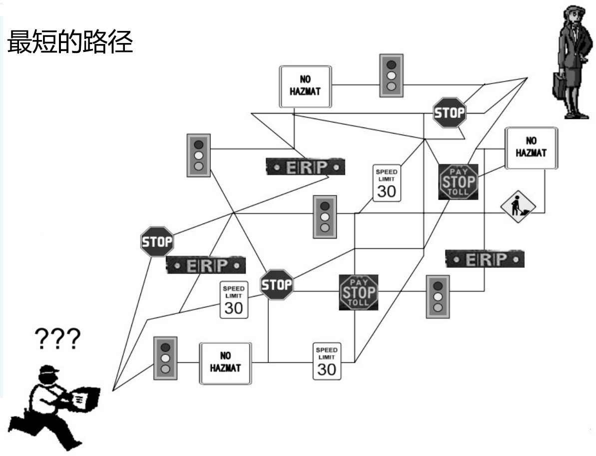
- 简介
- 数据处理：插值、拟合
- 数值积分
- 线性方程组的求解
- 非线性方程（组）的求解
- 常微分方程求解
- 最优化

最优化问题

- 最优化（规划）问题描述
- 最优化（规划）问题求解
  - 无约束最优化问题求解
  - 有约束最优化问题求解
  - 数值计算方法
  - Excel求解
  - Matlab求解

邮递员送信

- 最短的路径



最优化 (规划)

- 最优化方法是在第二次世界大战前后，在军事领域中对导弹、雷达控制的研究中逐渐发展起来的。
- 为了使系统达到最优的目标所提出的各种求解方法，这个过程就是规划，又称为最优化，是运筹学的一个重要分支。
- 最优化问题的应用遍布工业、社会、经济、管理等各个领域。
- 最优化方法已成为化学工程设计、项目论证、工艺变革及集成、经营管理等方面的一个重要手段。

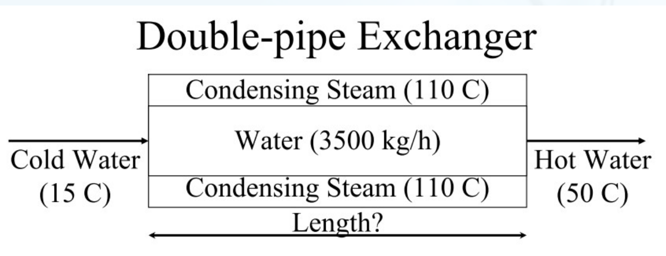
化工最优化问题

- 结构优化：流程方案的优化
  - 在多种流程方案中找到最优（费用最小、利润最大...），同时满足安全、环保、易操作等方面的要求。
- 参数优化：
  - 在给定流程结构条件下，优化化工过程系统的各项参数，如温度、压力、回流比等。

- 厂址选择
- 设备设计和工艺操作参数优化
- 过程优化控制
- 管道尺寸的确定和管线布置
- 维修周期和设备更新周期的确定
- 最优生产调度、最佳资源配置
- 原料和公用工厂的合理利用

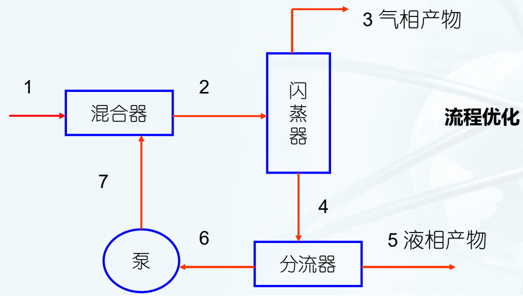
化工最优化问题

- 换热器长度最短?



化工最优化问题

下图为苯（**50%**）和甲苯（**50%**）混合物单级绝热闪蒸分离示意图，进料温度为**110° C**，压力为**182.4kPa**，试调整闪蒸压力和分流系数***s***，使气相产物中苯产量最大。



最优化方法数学描述

- 最优化：在一定的约束条件下，寻找使目标函数达到最优（最大、最小或者是特定目标值）的一组或者多组决策变量。
- 最优化问题标准数学形式：

目标函数： $\min\limits_{x\in R^n} J = F(x)$ **objective function**

约束条件 **restriction** $s.t.\quad G(x_i)=0,\quad i=1,2,3,...m_e$

$G(x_i)\le 0,\quad i=m_{e+1},...,m$

参数边界 $x_l \le x \le x_u$

决策变量 $X=\{x_1,x_2,\ldots,x_n\}$

若未知变量数等于独立等式约束方程数$\mathbf{m_e}$+独立不等式约束方程数$\mathbf{m_i}$,则不管优化准 则如何，至少存在一个解。

当约束条件中模型为非线性关系时，则存在多个解。

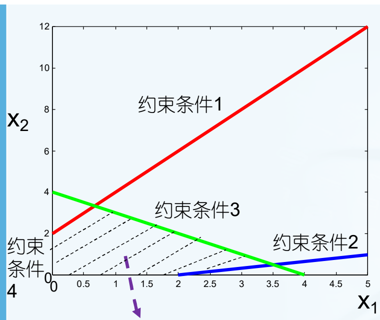
最优化问题基本概念

- 目标函数
  - 最优化问题均涉及到一个具体的最优目标，如产品收率最大、能耗最小、纯度要求、成本最低等。
  - 把目标写成数学形式的表达式称为目标函数。
- 约束条件与状态方程
  - 变量取值范围通常都有一定的限制，这种限制称为约束条件：
    - **不等式约束条件**，限制条件写成不等式形式
    - **等式约束条件**，以等式形式进行限定
  - 在化工过程的最优化问题中，物料衡算式、热平衡方程、动量守衡式均属等式约束，又称为系统或系统的状态方程。

最优化问题基本概念

- 决策变量和状态变量、系统自由度
  - 状态变量： 能描述系统的特征、行为的一组变量，其值应根据实际情况选取；
  - 决策变量： 由决策者根据目标或约束条件而确定操作变量或控制变量
    - 在化工系统中通常将能控制的变量如温度、压力、流量等做为决策变量。
  - 自由度： 在最优化问题中决策变量的个数称为自由度
- 确定系统的状态变量与决策变量时必须遵循的原则：
  - 状态变量数 = 状态方程数
  - 决策变量数 = 变量总数 - 状态变量数

最优化问题基本概念



可行域valid region
满足约束条件的点的全体集合

闭域与开域
若可行域包括边界上的所有点，则称此可行域为闭域，否则，称为开域。

- 极值：函数的极值只是局部的最优值，但却是全局最优值的侯选者。
- 对一元连续函数，存在极值点的
- 必要条件：$f'(X)=0$
- 充分条件：

$f''(X)>0$ (极小点)

$f''(X)<0$ (极大点)

最优化问题基本概念

对多元连续函数 $Y = f(X_1, X_1, \ldots\ldots, X_n)$

存在极值点的必要条件 $\nabla f(X) = \left(\frac{\partial f}{\partial X_1}, \frac{\partial f}{\partial X_2}, \ldots\ldots, \frac{\partial f}{\partial X_n}\right)^T = 0$

充分条件：

$$H = \begin{vmatrix} \frac{\partial^2 f}{\partial X_1^2} & \frac{\partial^2 f}{\partial X_1 \partial X_2} & \cdots & \frac{\partial^2 f}{\partial X_1 \partial X_n} \\ \frac{\partial^2 f}{\partial X_2 \partial X_1} & \frac{\partial^2 f}{\partial X_2^2} & \cdots & \frac{\partial^2 f}{\partial X_2 \partial X_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial X_n \partial X_1} & \frac{\partial^2 f}{\partial X_n \partial X_2} & \cdots & \frac{\partial^2 f}{\partial X_n^2} \end{vmatrix} \quad \text{（汉森矩阵）}$$

若H为正定（即各阶主子行列式的值均大于零）时，X^*为极小点；
若H为负定时（所有偶数行列式为正，所有奇数行列式为负），X^*为极大点；否则，X^*为鞍点。

最优化（规划）问题的分类

$\min \quad f(X)$ ----目标函数**objective function**

$subject \quad to \quad h_i(X) = 0 \quad i = 1 \sim m$ ----约束条件**restriction**

$g_j(X) \le 0 \quad j = 1 \sim n$

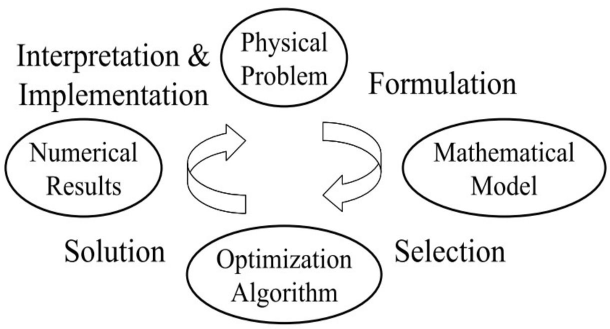
- 线性规划（Linear Programming，LP）：目标函数与约束均为线性
- 混合整数线性规划 (Mixed Integer Linear Programming, MILP)：线性规划模型中含有整数变量
- 非线性规划（Nonlinear Programming，NLP）：目标函数或约束条件其中一个含有非线性函数
- 混合整数非线性规划 (Mixed Integer Non-linear Programming, MINLP)：非线性规划中含有整数变量
- 二次规划（Quadratic programming）：目标函数为二次函数，约束条件为线性函数

最优化问题分类

- 无约束规划：不存在约束条件
- 有约束规划：存在约束条件
- 单目标优化：目标函数为一个
- 多目标优化：目标函数为多个
- 静态规划：与时间无关
- 动态规划：与时间有关

优化问题求解的步骤

Four Main Steps in Optimization



化工最优化（规划）问题求解

- 无约束最优化问题求解
  - 单变量无约束最优化问题
  - 多变量无约束最优化问题
- 有约束最优化求解
  - 线性规划、整数规划
  - 非线性规划、二次规划法
- 最小二乘法
- 多目标最优化

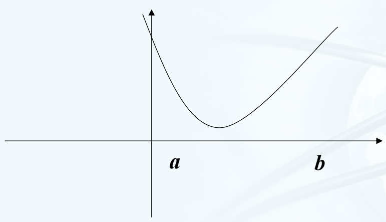
注：极大值问题可以转化为极小值问题来进行求解。

$$\max_x f(x) \Longrightarrow \min_x -f(x)$$

无约束最优化问题求解

单变量无约束优化问题

- **定义：** $\min_x J = f(x), \quad b < x < a$
- $f(x)$ **是区间** $[a,b]$ **上的单变量函数（一元函数）**



单变量无约束优化问题的求解

- 例：求函数$y=2x^3+3x^2-12x+14$在区间[-3,4]上的最大值与最小值。
- 解析法：导数=0
- 解：令$f(x)=y=2x^3+3x^2-12x+14$
- $f'(x)=6x^2+6x-12=6(x+2)(x-1)$
- 解方程$f'(x)=0$，得到$x_1=-2$，$x_2=1$，
- 由于f(-3)=23，f(-2)=34，f(1)=7，f(4)=142，
- 得：函数f(x)在x=4取得在[-3,4]上得最大值f(4)=142，在x=1处取得在[-3,4]上取得最小值f(1)=7

单变量无约束优化问题数值求解

- 求解单变量函数最优化方法又称一维搜索法。
- 基本思想：通过反复迭代，不断缩小搜索区间，最终求取函数近似极值点。
  - 1、选择初始点，尽量靠近最优解
  - 2、产生方向，使得 f(x) 从x_k出发,沿着方向$d^{(k)}$，可以找到x_{k+1}，有所下降。
  - 3、方向$d^{(k)}$确定后，求λ 使得f(x)下降最多，即求f(x(k)+λd(k))对λ的极值。
  - 4、检验新得到的迭代点是否满足要求。

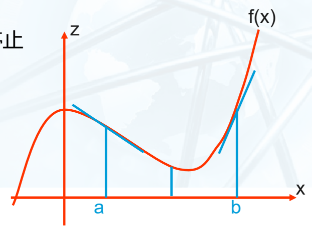
单变量无约束优化问题数值求解

- 一个好的搜索法应满足：
- 区间缩短率大；
- 选点方式简便；
- 对函数形式无特殊要求；
- 对初始点的选择要求不高。

- 一维搜索法：
- 对分法
- 黄金分割法 算法简单、稳定
- 二次多项式近似法(抛物线插值、二次插值)
- 三次多项式近似法

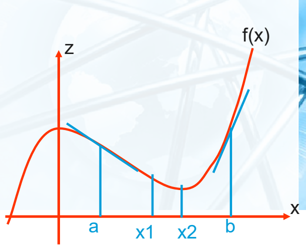
单变量无约束优化问题数值求解

- **对分法**
- 在区间[a,b]上，f'(a)<0, f'(b)>0,则在a， b之间必有f(x)的极小点，为了找到极小点，取试探点 X(k)=(a+b)/2,
- 若f'(X(k))>0,则取[a, X(k)]为新区间，若f'(X(k)) < 0,则取[X(k), b]为新区间.
- 直到 f'(X(k))充分小或区间充分小时停止
- 取当前区间的中点为极值点。



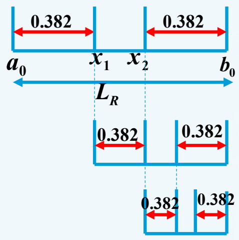
单变量无约束优化问题数值求解

- 黄金分割法(0.618法) Golden Section
  - 0.618法适用于单峰函数，即在所讨论的区间[a,b]上,函数有一个极小点
  - 0.618的意义：$x^2+x-1=0$ 的解（正根）
  - 在当前区间内选择两个试探点， x1， x2，x1< x2
  - 若 f(x1)> f(x2),则新区间取为[x1,b]
  - 若 f(x1)<= f(x2),则新区间取为[a, x2]
- 试探点的取法按照：
- $X^1=a+0.382(b-a)$
- $X^2=a+0.618(b-a)$
- 直到区间长度小于预设值，
- 区间内任意点均可作为所求
- 极小点的近似。



黄金分割法

- 从原始搜索区间（a_0,b_0）的每一端距离0.382处选两点 x_1、x_2，除开始需选两点外，以后每次选一个即可。



$$x_1 = a_0 + 0.382(b_0 - a_0)$$

$$x_2 = a_0 + 0.618(b_0 - a_0)$$

区间缩短率 $E_n = \lambda^{n-1} \quad (n \geq 2)$

$$\left(\lambda = \frac{\sqrt{5}-1}{2} = 0.618\right)$$

0.618具有如下性质：

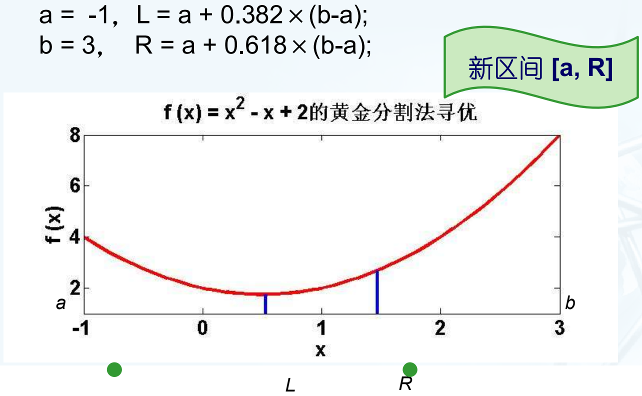
$0.618 \times 0.618 = 0.382$

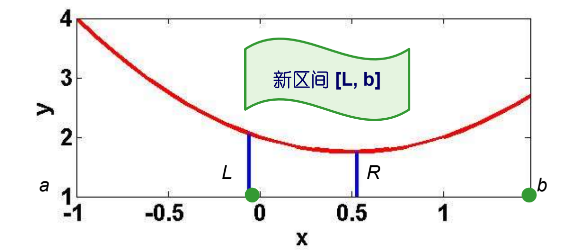
$0.382 + 0.382 \times 0.618 = 0.618$

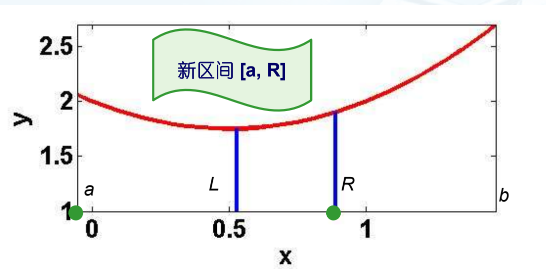
例 黄金分割法求 **f (x) = x^2 - x + 2**的极小点，初始区间为 **[-1, 3]**，要求迭代三步。

a = -1， L = a + 0.382 × (b-a);

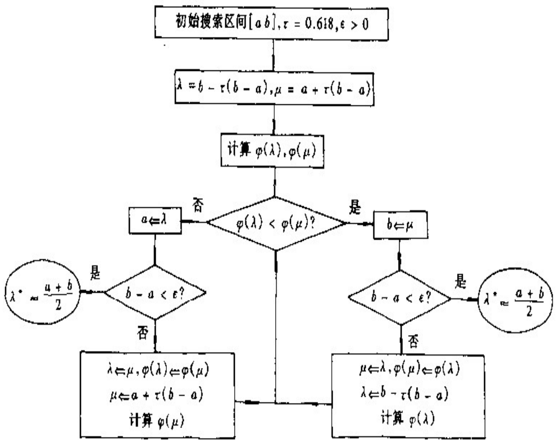
b = 3， R = a + 0.618 × (b-a);







黄金分割法流程图



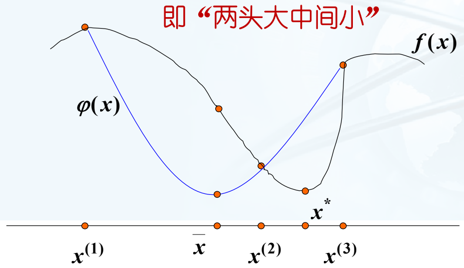
二次多项式近似（抛物线插值、二次插值）Parabolic interpolation

在极小点附近，用二次三项式 $\varphi(x)$ 逼近目标函数 $f(x)$，

令 $\varphi(x)$ 与 $f(x)$ 在三点 $x^{(1)} < x^{(2)} < x^{(3)}$ 处有相同的函数值，

并假设 $f(x^{(1)}) > f(x^{(2)}), \quad f(x^{(2)}) < f(x^{(3)})$.

即“两头大中间小”



如何计算函数 $\varphi(x)$？

设 $\varphi(x) = a + bx + cx^2$,

令
$$\varphi(x^{(1)}) = a + bx^{(1)} + cx^{(1)^2} = f(x^{(1)})$$
$$\varphi(x^{(2)}) = a + bx^{(2)} + cx^{(2)^2} = f(x^{(2)})$$
$$\varphi(x^{(3)}) = a + bx^{(3)} + cx^{(3)^2} = f(x^{(3)})$$

解上述方程组，可得逼近函数 $\varphi(x)$ 的系数 a, b 和 c.

再求函数 $\varphi(x)$ 的极小点， 令

$$\varphi'(x) = b + 2cx = 0,$$

解得 $\bar{x} = -\dfrac{b}{2c}$

$$= \frac{(x^{(2)^2} - x^{(3)^2})f(x^{(1)}) + (x^{(3)^2} - x^{(1)^2})f(x^{(2)}) + (x^{(1)^2} - x^{(2)^2})f(x^{(3)})}{2[(x^{(2)} - x^{(3)})f(x^{(1)}) + (x^{(3)} - x^{(1)})f(x^{(2)}) + (x^{(1)} - x^{(2)})f(x^{(3)})]}$$

以 $\bar{x}$ 作为 $f(x)$ 的极小点的估计值 。

抛物线插值算法步骤：

(1) 给定初始区间 $[x^{(1)}, x^{(3)}]$，设 $f(x^{(1)}) > f(x^{(2)}), \quad f(x^{(2)}) < f(x^{(3)})$.

令 $k := 1$。$\bar{x}^{(0)} = x^{(1)}$。给定精度 ε。

(2) 设 $\varphi(x) = a + bx + cx^2$。令 $\varphi(x^{(i)}) = f(x^{(i)}), i = 1, 2, 3$。解出

系数 a, b, c。解出 $\varphi(x)$ 的极小点 $\bar{x}^{(k)} = -\dfrac{b}{2c}$。

若 $|f(\bar{x}^{(k)}) - f(\bar{x}^{(k-1)})| < \varepsilon$，则算法停止，否则转（3）。

(3) 从 $x^{(2)}$ 和 $\bar{x}^{(k)}$ 中选择 $f(x)$ 的函数值最小的点及其

左、右两点，重新标记为 $x^{(1)}, x^{(2)}, x^{(3)}$。令 $k := k + 1$。转（2）。

单变量无约束优化问题Mworks求解：fminbnd

- 单变量（一元）函数无约束优化： $\min f(x), \quad x_1 < x < x_2$
- 函数fminbnd
- 函数库: TyOptimization
- 查找单变量函数在定区间上的最小值
- 算法基于黄金分割搜索和抛物线插值方法。
- 要求目标函数必须是连续函数，并可能只给出局部最优解.

- 常用格式如下:
- x,fval,exitflag,output = fminbnd(fun,x1,x2)

- 返回一个值 x，该值是 fun 中描述的标量值函数在区间 x1 < x < x2 中的局部最小值;
- 目标函数在 fun 的解 x 处计算出的值 fval;
- 描述退出条件的值 exitflag以及一个包含有关优化的信息的结构体 output。

单变量无约束优化问题Mworks求解

- 例：求 $f=2e^{-x}\sin x$ 在 $0<x<8$ 中的最小值与最大值.

```
f=x->2*exp(-x)*sin(x)
x=fminbnd(f,0,8)
--------------------------------------------
f1=x->-2*exp(-x)*sin(x)
x=fminbnd(f1,0,8)
```

运行结果：

xmin = 3.9270　　ymin = -0.0279
xmax = 0.7854　　ymax = 0.6448

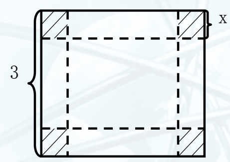
例　有边长为3m的正方形铁板，在四个角剪去相等的正方形以制成方形无盖水槽，问如何剪法使水槽的容积最大？

设剪去的正方形的边长为 x**，则水槽的容积为：** $(3-2x)^2x$

建立无约束优化模型为： $\min\ y=-(3-2x)^2x$，　$0<x<1.5$

先编写函数如下:

```
function tank(x)
  F=-(3-2*x).^2*x;
  return F
end
```



主程序:

```
x,fval,=fminbnd(tank,0,1.5);
xmax=x
fmax=-fval
```

运算结果为: xmax = 0.5000,fmax =2.0000.
剪掉的正方形的边长为0.5m时水槽的容积最大,最大容积为2m³.

多变量无约束优化问题

数学描述：$\min\limits_{x} J=f(x)$

$$x=[x_1\quad x_2\quad \cdots\quad x_{n-1}\quad x_n]\in R^n$$

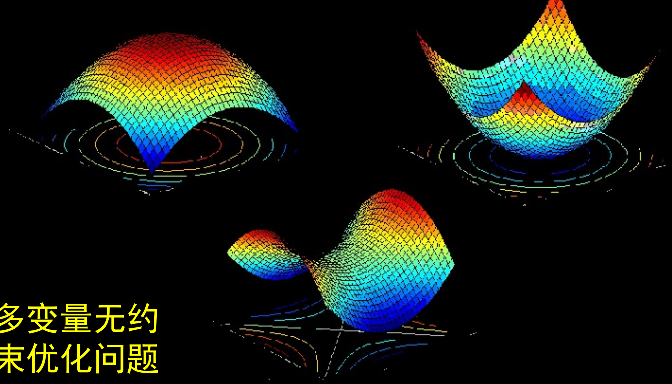
$$b_i<x_i<a_i$$

x包含多个变量，f(x)为多元函数

如：$f(x)=x_1^2+x_2^2$

$f(x_1\ \ x_2)=-x_1^2-x_2^2$

$f(x_1\ \ x_2)=x_1^2+x_2^2$



多变量无约束优化问题

$f(x_1\ \ x_2)=x_1^2-x_2^2$

多变量无约束优化问题数值求解

- 最速下降法steepest descent (梯度下降法 Gradient Descent)
- 共轭梯度法 Conjugate gradient method
- 牛顿法 Newton
- 单纯形法 Simplex

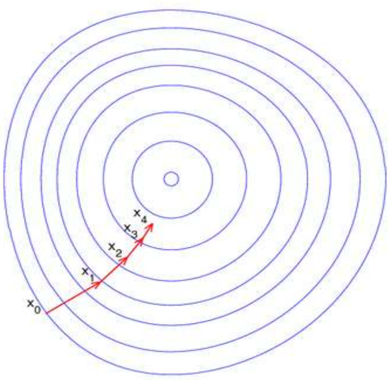
最速下降法

- 最速下降法又称为梯度法，是1847年由著名数学家Cauchy给出的，它是解析法中最古老的一种，其他解析方法或是它的变形，或是受它的启发而得到的，因此它是最优化方法的基础。
- 作为一种基本的算法，它在最优化方法中占有重要地位。其优点是工作量少，存储变量较少，初始点要求不高；缺点是收敛慢，效率不高，有时达不到最优解。
- 它的理论和方法渗透到许多方面，特别是在军事、经济、管理、生产过程自动化、工程设计和产品优化设计等方面都有着重要的应用。

最速下降法 $\min \quad y = f(\boldsymbol{x}) \quad \boldsymbol{x} \in R^n$

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T \cdot (\mathbf{x} - \mathbf{x}^k)$$

沿着负（正）梯度方向，函数值下降（或上升）最大，即负（正）梯度方向函数值下降（上升）最快，故取负（正）梯度方向为搜索方向



最速下降法 $\min \quad y = f(\boldsymbol{x}) \quad \boldsymbol{x} \in R^n$

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T \cdot (\boldsymbol{x} - \boldsymbol{x}^k)$$

$$\boldsymbol{x} - \boldsymbol{x}^k = -\nabla f(\boldsymbol{x}^k) \Rightarrow \boldsymbol{x} = \boldsymbol{x}^k - \nabla f(\boldsymbol{x}^k)$$

改进： $\boldsymbol{x} = \boldsymbol{x}^k - t^k \cdot \nabla f(\boldsymbol{x}^k)$ **步长**

最优步长的确定 $\min \quad g(t^k) = f\left[\boldsymbol{x} - t^k \cdot \nabla f(\boldsymbol{x}^k)\right]$

步骤：

a. 令迭代初值 $\boldsymbol{x}^0$ 为确定性演化的个体，$k = 0$，迭代上限 N；

b. 计算导数 $\nabla f(\boldsymbol{x}^k)$，若 $\left\|\nabla f(\boldsymbol{x}^k)\right\| \le \varepsilon$，则输出结果，停止计算；

c. 求解 $\min \quad g(t_k) = f\left\{\boldsymbol{x}^k - t_k \cdot \nabla f(\boldsymbol{x}^k)\right\}$，获得最优步长 t_k；

d. $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + t_k \cdot (-\nabla f(\boldsymbol{x}^k))$；

e. $k = k+1$，若 $k < N$，回到 b，否则输出结果，停止计算。

最速下降法求解化工过程最优化问题实例

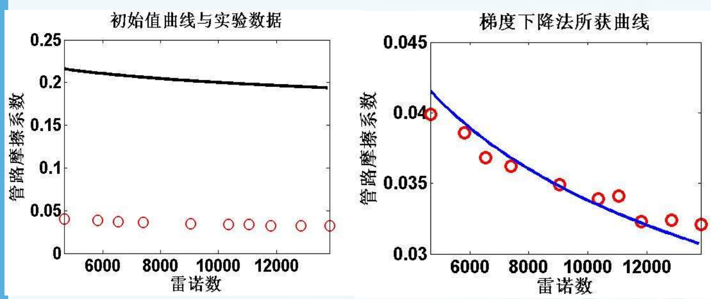
- 已知管路摩擦系数λ与雷诺数Re之间函数形式为下式所示，且根据实验获得如表所示数据，
- 要求获得下式中的公式参数a、b和c

$$\lambda = a \cdot \mathrm{Re}^b + c \qquad \min_{a,b,c} \; sse = \sum_{i=1}^{10} \left(a \cdot (\mathrm{Re}_i)^b + c - \lambda_i \right)^2$$

- 无约束非线性规划

Re	λ	Re	λ
4658	0.0399	10350	0.0339
5820	0.0386	11050	0.0341
6525	0.0368	11820	0.0323
7400	0.0362	12850	0.0324
9045	0.0349	13840	0.0321

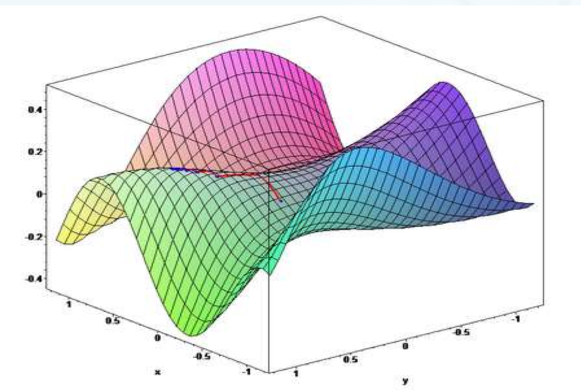
$$[a \quad b \quad c]_0 = [0.5 \quad -0.1 \quad 0.001] \qquad [a \quad b \quad c] = [0.2454 \quad -0.1 \quad -0.0639]$$



耗时2s，迭代869次，最终 sse = 9.9815e-006。
梯度下降法寻优速度较慢

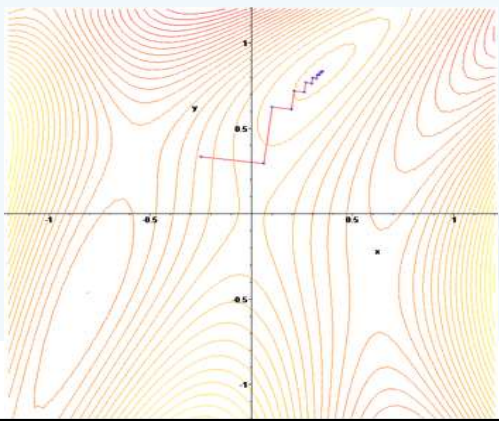
› 用最速下降法求此函数的最小值

$$F(x,y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$$



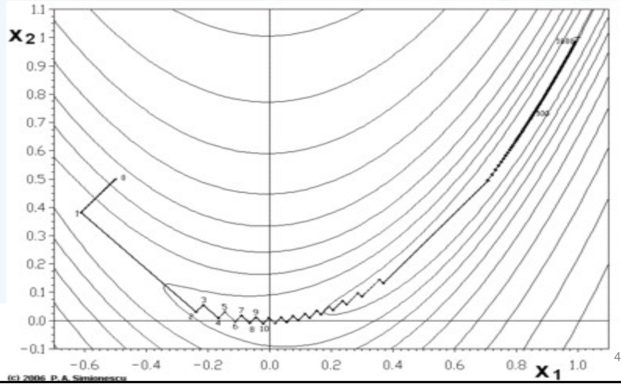
› 用最速下降法求此函数的最小值

$$F(x,y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$$



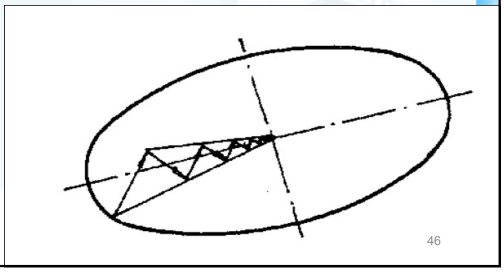
最速下降法处理一些复杂的非线性函数会出现问题，例如**Rosenbrock函数** $f(x,y)=(1-x)^2+100(y-x^2)^2$

该函数具有狭窄弯曲的山谷，最小值就在这些山谷之中，谷底很平。其最小值在（1,1）处，数值为f(1,1)=0。
此函数优化过程是之字形的向极小值点靠近，速度非常缓慢。



最速下降法

- 最速下降法注意的问题：
- 化工中有些问题不能或很难写出变量之间关系的明确表达式，此时可以用差分来逼近偏导数向量的各分量；
- 锯齿现象，‘之字型’下降。
- 靠近极小值时速度减慢。
- 直线搜索可能会产生一些问题。



多变量无约束优化问题

- 牛顿法
  - 思想：用二阶泰勒多项式近似目标函数，极值点处导数为0
  - 用近似多项式的极值点代替目标函数的极值点，得到一个点列。

$f(x^{(k+1)})=f(x^{(k)})+f'(x^{(k)})(x-x^{(k)})+0.5f''(x^{(k)})(x-x^{(k)})^2$

$f'(x^{(k+1)})=f'(x^{(k)})+f''(x^{(k)})(x-x^{(k)})=0$

$x^{(k+1)}=x^{(k)}-f'(x^{(k)})/f''(x^{(k)})$

  - 直到$f'(x^{(k)})$充分小。

牛顿法

- 将待优化函数在某点附近做二阶Taylor展开：

$$f(\boldsymbol{x})\approx Q(\boldsymbol{x})=f(\boldsymbol{x}^k)+\nabla f(\boldsymbol{x}^k)^T\cdot(\boldsymbol{x}-\boldsymbol{x}^k)+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^k)^T\cdot\boldsymbol{H}(\boldsymbol{x}^k)\cdot(\boldsymbol{x}-\boldsymbol{x}^k)$$

则Q (x)的最优点求解为：

牛顿法搜索方向

$$\nabla Q(\boldsymbol{x}^*)=\boldsymbol{0} \Rightarrow \boldsymbol{x}^*=\boldsymbol{x}^k\boxed{-H(\boldsymbol{x}^k)^{-1}\cdot\nabla f(\boldsymbol{x}^k)}$$

牛顿法迭代公式：

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-H(\boldsymbol{x}^k)^{-1}\cdot\nabla f(\boldsymbol{x}^k)$$

下山单纯形法

- Nelder-Mead法或称下山单纯形法，由Nelder和Mead发现（1965年），这是用于优化多维无约束问题的一种数值方法，属于更一般的搜索算法的类别。
- 单纯形是N维中的N + 1个顶点的凸包，是一个多胞体
- 单纯形是空间中最简单的图形，如直线上的一个线段，平面上的一个三角形，三维空间中的一个四面体，等等。
- 单纯形求解无约束优化问题的基本思路：
  - 计算单纯形顶点的函数值，通过比较它们的大小来判别极值点的搜索方向。
  - 用不断更新单纯形的方法，使单纯形的某个顶点逼近极值点，当达到计算精度要求时，迭代结束。

多变量无约束优化问题 Mworks求解

- 多变量无约束优化问题求解函数fminunc： $\min_x f(x)$
- 函数库: TyOptimization
- 非线性规划求解器,该函数采用梯度法和牛顿法

x,fval,exitflag,output,grad,hessian = fminunc(fun,x0)

- 在点 x0 处开始并尝试求 fun 中描述的函数的局部最小值 x，x0 为向量。
- fminunc 返回以下值：
  - x - 函数的局部最小值。
  - fval - 目标函数 fun 在解 x 处的值。
  - exitflag - 描述 fminunc 的退出条件的值。
  - output - 提供优化过程信息的结构体。
  - grad - fun 在解 x 处的梯度。
  - hessian - fun 在解 x 处的 Hessian 矩阵。

多变量无约束优化问题Mworks求解

- 多变量无约束优化问题求解函数：fminsearch $\min_x f(x)$
- 函数库: TyOptimization
- 使用无导数法计算无约束的多变量函数的最小值
- 非线性规划求解器，该函数采用单纯形法

- x,fval,exitflag,output = fminsearch(fun,x0)
- 在点 x0 处开始并尝试求 fun 中描述的函数的局部最小值 x
- 在 fval 中返回目标函数 fun 在解 x 处的值
- 描述退出条件的值 exitflag
- 包含有关优化过程的信息的结构体 output。

例 用fminsearch函数求解

$$f(x,y)=(1-x)^2+100(y-x^2)^2$$

输入命令:

```
f=x-> (1-x[1])^2+100*(x[2]-x[1]^2)^2;  #x=[x,y]
x,fval,=fminsearch(f,[-1.2,2])
```

运行结果:

```
x =1.0000    1.0000
fval =1.9151e-010
```

- 练习：用mworks求解下列函数的最小值

$$\min \ f(\boldsymbol{x})=x_1^2+25x_2^2$$

有约束最优化问题

有约束最优化（规划）问题

$$\min \quad f(X)$$ ----目标函数**objective function**

$$subject \quad to \quad h_i(X)=0 \quad i=1\sim m$$
$$g_j(X)\le 0 \quad j=1\sim n$$
----约束条件**restriction**

- 线性规划（Linear Programming）：目标函数与约束均为线性
- 整数规划（Integer Programming）：最优化问题中的所有变量均为整数
- 混合整数线性规划 (Mixed Integer Linear Programming, MILP)：线性规划模型中含有整数变量
- 非线性规划（Nonlinear Programming）：目标函数或约束条件其中一个含有非线性函数
- 混合整数非线性规划 (Mixed Integer Non-linear Programming, MINLP)：非线性规划中含有整数变量
- 二次规划（Quadratic programming）：目标函数为二次函数，约束条件为线性函数

线性规划

$$\max or \min \ J=c_1x_1+c_2x_2+\ldots\ldots+c_nx_n$$

$$s.t.\quad \left.\begin{array}{l} a_{11}x_1+a_{12}x_2+\ldots\ldots+a_{1n}x_n\le b_1(\text{或}\ge b_1)\\ a_{21}x_1+a_{22}x_2+\ldots\ldots+a_{2n}x_n\le b_2(\text{或}\ge b_2)\\ \ldots\ldots\\ a_{m1}x_1+a_{m2}x_2+\ldots\ldots+a_{mn}x_n\le b_m(\text{或}\ge b_m)\end{array}\right\}$$ 约束方程组为线性

$$x_j\ge 0(j=1,\ldots\ldots,n)$$ 所有变量为非负值

- 注：
- 初始约束条件为等式的-------标准型初始模型；
- 初始约束条件有等式的和“≤”的--------规范型初始模型；
- 初始约束条件中有“≥”的--------一般型初始模型；

问题： 某厂每日8小时的产量不低于1800件.为了进行质量控制，计划聘请两种不同水平的检验员.一级检验员的标准为：速度25件/小时，正确率98%，计时工资4元/小时；二级检验员的标准为：速度15件/小时，正确率95%，计时工资3元/小时.检验员每错检一次，工厂要损失2元.为使总检验费用最省，该工厂应聘一级、二级检验员各几名？

解：设需要一级和二级检验员的人数分别为x_1、x_2人，

则应付检验员的工资为：$8\times4\times x_1+8\times3\times x_2=32x_1+24x_2$

因检验员错检而造成的损失为：

$$(8\times25\times2\%\times x_1+8\times15\times5\%\times x_2)\times2=8x_1+12x_2$$

目标函数：$\min z=(32x_1+24x_2)+(8x_1+12x_2)=40x_1+36x_2$

约束条件：$\begin{cases}8\times25\times x_1+8\times15\times x_2\ge1800\\ x_1,x_2\ge0\end{cases}$

运　输　问　题

要从甲城调出蔬菜2000吨，从乙城调出蔬菜2500吨，从丙地调出3000吨，分别供应A地2000吨，B地2300吨、C地1800吨、D地1400吨，已知每吨运费如下表：

供应单位 / 调出单位	A	B	C	D
甲	21	27	13	40
乙	45	51	37	20
丙	32	35	20	30

问：如何调拨才能使运费最省？

假设：
①假设题目中所给运费已考虑各地间公里数；
②只考虑运量和运费，不考虑车辆调拨等其它相关因素
③不考虑车辆返空的费用（或：所给运费已包含车辆返空的费用）
变量说明：
x_{ij}:从第i城运往第j地的蔬菜数量（i=1,2,3;j=1,2,3,4)
a_{ij}:从第i城运往第j地的单位运费（ i=1,2,3;j=1,2,3,4 ）
b_i:从第i城调出的蔬菜总量
c_j:第j地所需蔬菜总量

$$\min \quad z=\sum_{i=1}^{3}\sum_{j=1}^{4}a_{ij}x_{ij} \quad \text{s.t.} \begin{cases} \sum_{j=1}^{4}x_{ij}=b_i & (i=1,2,3) \\ \sum_{i=1}^{3}x_{ij}=c_j & (j=1,2,3,4) \\ x_{ij}\ge 0 & \\ x_{ij}\le \min(b_i,c_j) & (i=1,2,3;\ j=1,2,3,4) \end{cases}$$

整数规划

- 最优化问题中的所有变量均为整数时，这类问题称为整数规划问题。
- 如果线性规划中的所有变量均为整数时，称这类问题为线性整数规划问题。
- 整数规划可分为线性整数规划和非线性整数规划，以及混合整数规划等。
- 如果决策变量的取值要么为0，要么为1，则这样的规划问题称为0 - 1规划。

引例.资源分配问题：

某个中型的百货商场要求售货人员每周工作5天，连续休息2天，工资200元/周，已知对售货人员的需求经过统计分析如下表，问如何安排可使配备销售人员的总费用最少？

星期	一	二	三	四	五	六	日
所需售货员人数	18	15	12	16	19	14	12
开始休息的人数	x_1	x_2	x_3	x_4	x_5	x_6	x_7

设决策变量如上，可建立如下模型：

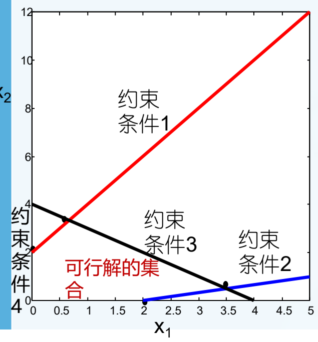
$$\min \quad z = 200(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7)$$

$$s.t.\begin{cases} x_2 + x_3 + x_4 + x_5 + x_6 \ge 18 \\ x_3 + x_4 + x_5 + x_6 + x_7 \ge 15 \\ x_4 + x_5 + x_6 + x_7 + x_1 \ge 12 \\ x_5 + x_6 + x_7 + x_1 + x_2 \ge 16 \\ x_6 + x_7 + x_1 + x_2 + x_3 \ge 19 \\ x_7 + x_1 + x_2 + x_3 + x_4 \ge 14 \\ x_1 + x_2 + x_3 + x_4 + x_5 \ge 14 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 \ge 0 \text{且为整数} \end{cases}$$

61

单纯形法（simplex algorithm）

George Dantzig发明的单纯形法（simplex algorithm）是线性规划问题的数值求解的流行技术。

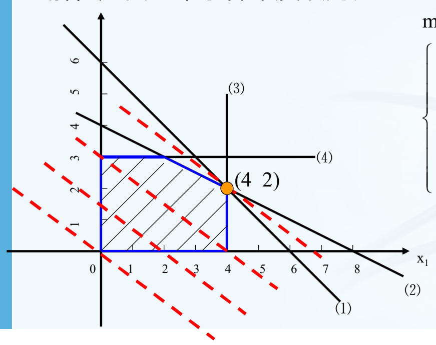


- 单纯形: 是N维中的N+1个顶点的凸包，是一个多胞体（直线上的一个线段，平面上的一个三角形，三维空间中的一个四面体，等等）
- 基本思路：先求得基本可行解，然后在其中找到最优。
- 几何意义：在凸集合的顶点之间进行变换，找到最优。
- 单纯形法并不需要检测每一个顶点才能找到最优解
- 可以证明，一般计算次数在m~2m之间（m--约束方程个数）

62

线性规划求解图解法

- 例：建立直角坐标，图中阴影部分及边界上的点均为其解，是由约束条件来反映的。



$$\max \quad Z = 2x_1 + 3x_2$$

$$\begin{cases} 2x_1 + 2x_2 \le 12 & (1) \\ x_1 + 2x_2 \le 8 & (2) \\ 4x_1 \le 16 & (3) \\ 4x_2 \le 12 & (4) \\ x_1 \ge 0,\ x_2 \ge 0 \end{cases}$$

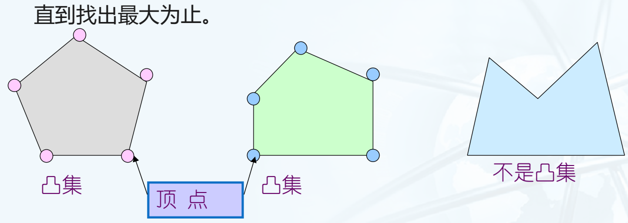
最优解：$x_1 = 4$ $x_2 = 2$

有唯一最优解，$Z = 14$

63

解的基本定理

- 线性规划问题的可行域是凸集（凸多边形）。
- 最优解一定是在凸集的某一顶点实现（顶点数目不超过 C_n^m 个）
- 先找一个基本可行解，与周围顶点比较，如不是最大，继续比较，直到找出最大为止。



解的情况 $\begin{cases}\text{唯一解} \\ \text{无穷解} \\ \text{无界解} \\ \text{无可行解}\end{cases}$

64

Matlab求解线性规划问题

$$\min_x f^T x \text{使得} \begin{cases} A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub \end{cases}$$

f、x、b、beq、lb 和 ub 是向量，A 和 Aeq 是矩阵。

- linprog函数：线性规划求解器
- 函数库: TyOptimization
- 函数使用形式:
- x,fval,exitflag,output,lambda = linprog(f,A,b,Aeq,beq,lb,ub)

- 注意：f是系数向量，指定为实数向量。
- 系数向量表示目标函数 f'*x。该表示法假设 f 是列向量。

- 该函数返回值为:
- x - 线性规划的解
- fval - 目标函数在解 x 处的值：fval = f'*x
- exitflag - 说明退出条件的值
- output - 包含优化过程信息的结构体
- lambda - 在解 x 处的拉格朗日乘数

Mworks求解线性规划问题

- 例：假设某厂有原料M_1=60kg，M_2=100kg，M_3=60kg，可生产P_1、P_2两种产品。生产1kg P_1需$M_1$2kg、$M_2$2kg，生产1kg P_2需$M_1$1kg、$M_2$5kg、$M_3$4kg。
- 销售1kg P_1的收入为6元，销售1kg P_2的收入为7元。问如何安排生产计划，可使收入最大
- 解：设P_1的产量为$x_1$kg，P_2的产量为$x_2$kg

- f=[-6,-7];A=[2 1;2 5;0 4];b=[60,100,60];
- x, favl, = linprog(f,A,b,[],[],[0,0],[])

$$\begin{aligned} \max \quad & y = 6x_1 + 7x_2 \\ S.t. \quad & 2x_1 + x_2 \le 60 \\ & 2x_1 + 5x_2 \le 100 \\ & 4x_2 \le 60 \\ & x_j \ge 0, j = 1 \sim 2 \end{aligned}$$

- 求得最优解:
- 生产 P_1 25kg，P_2 10kg，
- 最大收入220元。

非线性规划

- 如果目标函数或约束条件中至少有一个是非线性函数时的最优化问题就叫做非线性规划问题

$$\min \ f(X)$$
$$s.t. \begin{cases} g_i(X) \ge 0 & i = 1,2,\ldots m; \\ h_j(X) = 0 & j = 1,2,\ldots l. \end{cases}$$
$$X = (x_1, x_2, \cdots, x_n)^T$$

其它情况: 求目标函数的最大值或约束条件为小于等于零的情况，都可通过取其相反数化为上述一般形式.

有约束非线性规划求解方法

- 将复杂的非线性优化问题转为较简单的线性规划问题或二次规划问题:
- 序列二次规划法

- 将有约束极值问题转化为无约束极值问题:
- 拉格朗日乘子法和罚函数法

建立模型

记工地的位置为(a_i，b_i)，水泥日用量为d_i，i=1,…,6;料场位置为(x_j，y_j)，日储量为e_j，j=1,2；料场j向工地i的运送量为X_{ij}。

目标函数为：$\min f=\sum_{j=1}^{2}\sum_{i=1}^{6}X_{ij}\sqrt{(x_j-a_i)^2+(y_j-b_i)^2}$

约束条件为：

$$\sum_{j=1}^{2}X_{ij}=d_i,\quad i=1,2,\cdots,6$$

$$\sum_{i=1}^{6}X_{ij}\le e_j,\quad j=1,2$$

当用临时料场时决策变量为：X_{ij}，

当不用临时料场时决策变量为：X_{ij}，x_j，y_j。

Mworks求解二次规划问题

标准型为：　Min Z= $\frac{1}{2}X^THX+c^TX$

s.t. $AX\le b$　$Aeq\cdot X=beq$　$VLB\le X\le VUB$

Mworks二次规划函数quadprog

具有线性约束的二次目标函数的求解器，仅适用于基于求解器的方法。

函数库: TyOptimization

x,fval,exitflag,output,lambda = quadprog(H,f,A,b,Aeq,beq,lb,ub)

要使问题具有有限最小值，输入 H 必须为正定矩阵。

如果 H 是正定矩阵，则解 x = H\(-f)。

该函数返回值为：

x - 解向量；

fval - 在 x 处的目标函数值：fval = 0.5*x'*H*x + f'*x；

exitflag - 描述 quadprog 退出条件的整数；

output - 有关优化信息的结构体；

lambda - 在解 x 处的拉格朗日乘数。

例1　min $f(x_1,x_2)=-2x_1-6x_2+x_1^2-2x_1x_2+2x_2^2$

s.t. $x_1+x_2\le 2$

$-x_1+2x_2\le 2$

$x_1\ge 0,\ x_2\ge 0$

1、写成标准形式：

$$\min z=(x_1,x_2)\begin{pmatrix}1&-1\\-1&2\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}+\begin{pmatrix}-2\\-6\end{pmatrix}^T\begin{pmatrix}x_1\\x_2\end{pmatrix}$$

$$\text{s.t.}\ \begin{pmatrix}1&1\\-1&2\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}\le\begin{pmatrix}2\\2\end{pmatrix}$$

$$\begin{pmatrix}0\\0\end{pmatrix}\le\begin{pmatrix}x_1\\x_2\end{pmatrix}$$

2、 输入命令：

```
H=[1 -1; -1 2];
c=[-2,-6];A=[1 1; -1 2];b=[2,2];
Aeq=[];beq=[]; VLB=[0,0];VUB=[];
[x,z]=quadprog(H,c,A,b,Aeq,beq,VLB,VUB)
```

3、运算结果：　x =0.6667　1.3333　z = -8.2222

Mworks求解一般有约束非线性规划：fmincon

标准型为：　$\min F(X)$

$s.t\ AX\le b$，　$Aeq\cdot X=beq$，　$G(X)\le 0$，

$Ceq(X)=0$，　$VLB\le X\le VUB$.

- 求解函数fmincon寻找约束非线性多变量函数的最小值
- 函数库: TyOptimization
- x,fval,exitflag,output,lambda,grad,hessian =fmincon(fun,X0,A,b)
- x,...=fmincon(fun,X0,A,b,Aeq,beq,VLB,VUB,nonlcon)
- x,...=fmincon(fun,X0,A,b,Aeq,beq,VLB,VUB,nonlcon,options)
- 从 x0 开始，寻找 fun 中所述的函数的最小值点 x。x0 为向量。
- x-最小值
- fval - 标量，目标函数 fun 在解 x 处的值。
- exitflag - 整数，描述 fmincon 的退出条件的值。
- output - 结构体，提供优化过程信息。
- lambda - 结构体，其字段包含解 x 处的拉格朗日乘数。
- grad - fun 在解 x 处的梯度。
- hessian - fun 在解 x 处的黑塞矩阵。请参阅fmincon Hessian 矩阵。

Mworks求解一般有约束非线性规划：fmincon

- fmincon 是基于梯度的方法，旨在处理目标函数和约束函数均为连续且具有连续一阶导数的问题。
- fmincon函数可能会给出局部最优解，这与初值X_0的选取有关。

- 定义目标函数fun:
- --------格式一-------------
- fun = x -> ......
- --------格式二-------------
- fun = x-> begin
-     f = ...
-     return f
- end
- --------格式三--------------
- function fun(x)
-     f=....
-     return f
- end
- ---------------------

若有非线性约束: G(X) 或Ceq(X)
建立非线性约束函数nonlcon
定义函数G(X)与Ceq(X):

```
nonlcon = x -> begin
    G= ......
    Ceq = ......
    return c,ceq
end
```


例2 $\min\ f = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$

$$\text{s.t}\quad \begin{cases} 2x_1+3x_2 \le 6 \\ x_1+4x_2 \le 5 \\ x_1, x_2 \ge 0 \end{cases}$$

1、写成标准形式:

$$\min\ f = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$$

$$s.t.\quad \begin{pmatrix} 2x_1+3x_2-6 \\ x_1+4x_2-5 \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \le \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

2、先建立目标函数fun1:

fun1=x->-x[1]-2*x[2]+(1/2)*x[1]^2+(1/2)*x[2]^2

3、建立初始值、约束条件和主程序:

```
x0=[1,1];
A=[2 3; 1 4]; b=[6,5];
Aeq=[];beq=[];
VLB=[0,0]; VUB=[];
x,fval=fmincon(fun1,x0,A,b,Aeq,beq,VLB,VUB)
```

4、运算结果为:

x = 0.7647   1.0588
fval =   -2.0294

$$\min\ f = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$$

$$\begin{pmatrix} 2x_1+3x_2-6 \\ x_1+4x_2-5 \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \le \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

例3 $\mathrm{Min} f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$

$$\text{s.t.}\quad \begin{cases} x_1+x_2=0 \\ 1.5+x_1x_2 - x_1 - x_2 \le 0 \\ -x_1x_2-10 \le 0 \end{cases}$$

1．建立目标函数 fun2:

fun2=x-> exp(x[1])*(4*x[1]^2+2*x[2]^2+4*x[1]*x[2]+2*x[2]+1);

2．再定义非线性约束nonlcon:

```
nonlcon = x -> begin
    g=[1.5+x[1]*x[2]-x[1]-x[2],-x[1]*x[2]-10];
    ceq=[];
    return g,ceq
end
```

3．定义线性约束条件:

x0=[-1,1]; A=[];b=[]; Aeq=[1 1];beq=[0]; vlb=[];vub=[];

例3：fmincon

- 4-1．求解程序
- x,fval,=fmincon(fun2,x0,A,b,Aeq,beq,vlb,vub,nonlcon, options)
- 采用默认的"active-set" 算法，出现了bug，解决方法：尝试sqp算法

```
ERROR: ArgumentError: number of rows of each array must match (got (1, 3))
Stacktrace:
  [1] _typed_hcat(#unused#::Type{Float64}, A::Tuple{Matrix{Float64}, Matrix{Float64}})
    @ Base .\abstractarray.jl:1654
  [2] typed_hcat
    @ .\abstractarray.jl:1641 [inlined]
  [3] hcat
    @ .\array.jl:1956 [inlined]
  [4] _qrinsert(Q::LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}, Matrix{Float64}}, R::Matrix{Float64}, j::Int64, x::Vector{Float64})
    @ TyOptimCore.__Internal__
C:\Users\Public\TongYuan\.julia\packages\TyOptimCore\Phf9k\src\NonlinearOptimization\SolverBasedNonlinearOptimization\nlconst_utils.jl:23
  [5] _qpsub(H::Vector{Float64}, f::Vector{Float64}, A::Matrix{Float64}, B::Matrix{Float64}, lb::Vector{Float64}, ub::Vector{Float64}, X::Vector{Float64}, neqcstr::Int64, verbosity::Int64, caller::String, ncstr::Int64, nov::Int64, options::NamedTuple{(:MaxIter, :TolCon), Tuple{Int64, Float64}}, ACTIND::UnitRange{Int64}, phaseOneTotalScaling::Bool)
    @ TyOptimCore.__Internal__
C:\Users\Public\TongYuan\.julia\packages\TyOptimCore\Phf9k\src\NonlinearOptimization\SolverBasedNonlinearOptimization\nlconst.jl:1342
  [6] _qpsub
    @ C:\Users\Public\TongYuan\.julia\packages\TyOptimCore\Phf9k\src\NonlinearOptimization\SolverBasedNonlinearOptimization\nlconst.jl:851 [inlined]
  [7] _qpsub(H::Matrix{Float64}, f::Vector{Float64}, A::Adjoint{Float64, Matrix{Float64}}, B::Matrix{Float64}, lb::Vector{Float64}, ub::Vector{Float64}, X::Vector{Float64}, neqcstr::Int64, verbosity::Int64, caller::String, ncstr::Int64, nov::Int64, options::NamedTuple{(:MaxIter, :TolCon), Tuple{Int64, Float64}}, ACTIND::Vector{Int64}, phaseOneTotalScaling::Bool)
    @ TyOptimCore.__Internal__
C:\Users\Public\TongYuan\.julia\packages\TyOptimCore\Phf9k\src\NonlinearOptimization\SolverBasedNonlinearOptimization\nlconst.jl:1052
  [8] nlconst(funfcn::Tuple{String, String, var"#43#44", Nothing, Nothing}, x::Vector{Float64}, lb::Vector{Float64}, ub::Vector{Float64}, Ain::Matrix{Float64}, Bin::Vector{Float64}, Aeq::Mat
```


例3：fmincon

4-2．求解程序

options = optimoptions(:fmincon,Algorithm="sqp")

#尝试 "sqp" 算法，该算法有时比默认的 "active-set" 算法更快或更准确。

x,fval,=fmincon(fun2,x0,A,b,Aeq,beq,vlb,vub,nonlcon, options)

4. 运算结果为：

x = -1.2250　1.2250

fval = 1.8951

算法的选择

- options — 优化选项
- optimoptions 的输出
- 优化选项，指定为 optimoptions 的输出。
- 一些选项适用于所有算法，其他选项则与特定算法相关。

所有算法	
Algorithm	选择优化算法： • "trust-region-reflective" • "sqp" • "active-set"（默认值） trust-region-reflective 算法要求： • 在目标函数中提供梯度 • 将 SpecifyObjectiveGradient 设置为 true • 具有边界约束或线性等式约束之一，但不能两者都有 如果您选择 "trust-region-reflective" 算法，但这些条件无法全部得到满足，fmincon 会引发错误。"active-set" 和 "sqp" 算法不是大规模算法。

最小二乘法的mworks函数介绍

- lsqlin: 求解约束的线性最小二乘问题
- x=lsqlin(C,d,A,b,Aeq,beq,lb,ub,x0,options)
- 函数库: TyOptimization

求解以下形式的最小二乘曲线拟合问题：

$$\min_x \frac{1}{2}||C\cdot x-d||_2^2 \text{使得} \begin{cases} A\cdot x\le b \\ Aeq\cdot x = beq \\ lb\le x\le ub \end{cases}$$

- lsqnonnegm: 求解非负的线性最小二乘问题
- 函数库: TyOptimization
- x= lsqnonnegm(C, d, options)

求解以下形式的非负最小二乘曲线拟合问题

$$\min_x ||C\cdot x-d||_2^2, \text{其中} x\ge 0$$

最小二乘法的mworks函数介绍

- lsqnonlin: 求解非线性最小二乘问题
- 函数库: TyOptimization
- x = lsqnonlin(fun,x0,lb,ub,A,b,Aeq,beq,nonlcon,options)

求解以下形式的非线性最小二乘曲线拟合问题

$$\min_x \|f(x)\|_2^2 = \min_x (f_1(x)^2 + f_2(x)^2 + \ldots + f_n(x)^2)$$

需满足以下约束

```
lb≤x
x≤ub
Ax≤b
Aeqx=beq
c(x)≤0
ceq(x)=0
```

不要将目标函数指定为标量值 $\|f(x)\|_2^2$（平方和）

lsqnonlin 要求目标函数是向量值函数

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

Mworks单目标的适用求解器

•fmincon 应用于大多数具有平滑约束的平滑目标函数。它没有列为最小二乘、线性或二次规划的首选求解器，因为表中列出的对应求解器通常更高效。

约束类型	目标类型				
	线性	二次	最小二乘	平滑非线性	非平滑
无	不适用（f = const，或 min = −∞）	quadprog、信息	mldivide、lsqcurvefit、lsqnonlin、信息	fminsearch、fminunc、信息	fminsearch
边界	linprog、信息	quadprog、信息	lsqcurvefit、lsqlin、lsqnonlin、lsqnonnegm、信息	fminbnd、fmincon、fseminf、信息	fminbnd
线性	linprog、信息	quadprog、信息	lsqcurvefit、lsqlin、lsqnonlin、信息	fmincon、fseminf、信息	\
锥	\	fmincon、信息	lsqcurvefit、fmincon、lsqnonlin、信息	fmincon、信息	\
常规平滑	fmincon、信息	fmincon、信息	lsqcurvefit、fmincon、lsqnonlin、信息	fmincon、fseminf、信息	\
离散，具有边界或线性	intlinprog、信息	\	\	\	\

智能优化算法

- 1975年holland提出遗传算法 (Genetic Algorithm)
- 1977年Glouer提出禁忌搜索算法 (Tabn Search)
- 1982年Kirkpatrick提出模拟退火算法 (Simulated Annealing)
- 人工神经元网络
- 1995年Dorigo提出蚁群算法 (Ant Colony Optimization)
- 1995年Kennedy & Eherhart提出粒子群优化 (Particle Swarm Optimization)
- 其它
- 文化算法 (Cultural Algorithm)
- 人工生命算法 (Artificial-Life Algorithm)
- 我们统称以上算法为人工生命计算 (Artificial Life Computation)
- 人工生命计算 + 模糊逻辑 (Fuzzy Logic)=软计算(Soft Computation)
- 人工生命计算 + 进化编程 = 进化算法 (Evolutionary computation)
- 启发式、智能计算、进化计算、自然计算

练习：用Excel规划求解器求解

- 例 某机床厂生产甲、乙两种机床，每台销售后的利润分别为4000元与3000元。生产甲机床需用机器加工，加工时间分别为每台2小时和1小时；生产乙机床需用三种机器加工，加工时间为每台各一小时。若每天可用于加工的机器时数分别为机器10小时、机器8小时和机器7小时，问该厂应生产甲、乙机床各几台，才能使总利润最大？
- 设该厂生产x1台甲机床和x2乙机床
- 上述问题的数学模型：　目标函数 $\max \quad z = 4000x_1 + 3000x_2$
- 约束条件 s.t. $\begin{cases} 2x_1 + x_2 \le 10 \\ x_1 + x_2 \le 8 \\ x_2 \le 7 \\ x_1, x_2 \ge 0 \end{cases}$

练习：用Excel规划求解器求解

例某厂计划在下一个生产周期内生产甲、乙两种产品，已知资料如表所示。试制定生产计划，使获得的利润最大？同时，根据市场预测，甲的销路不是太好，应尽可能少生产；乙的销路较好，可以扩大生产。试建立此问题的数学模型。

单位消耗 产品 资源	甲	乙	资源限制
钢材	9	4	3600
煤炭	4	5	2000
设备台时	3	10	3000
单件利润	70	120	

（1） $\max Z=70x_1+120x_2$

$9x_1+4x_2\le 3600$

$4x_1+5x_2\le 2000$

$3x_1+10x_2\le 3000$

$x_1,\ x_2\ge 0$

（2） $\max Z_1=70x_1+120x_2$

$\max Z_2=x_1$

$\max Z_3=x_2$

$9x_1+4x_2\le 3600$

$4x_1+5x_2\le 2000$

$3x_1+10x_2\le 3000$

$x_1,\ x_2\ge 0$

小结

- 了解最优化问题
- 学会求解无约束最优化问题
- 学会求解有约束最优化问题
- 重点：
- 有约束最优化问题的mworks函数：fmincon
- Excel规划求解器求解线性规划
- 了解人工智能及其优化算法