数值计算方法 期末课程大作业

精馏过程中的优化生产问题

浙江大学·控制系

赵聪

学号：3071001269

目录

问题背景

精馏是化学工业中一种重要的单元操作，在广泛的行业中有重要的工业运用，它也是我们《化工原理》课程中的重要组成部分。精馏操作按不同方法进行分类。根据操作方式，可分为连续精馏和间歇精馏；根据混合物的组分数，可分为二元精馏和多元精馏；根据是否在混合物中加入影响汽液平衡的添加剂，可分为普通精馏和特殊精馏（包括萃取精馏、恒沸精馏和加盐精馏）。若精馏过程伴有化学反应，则称为反应精馏。

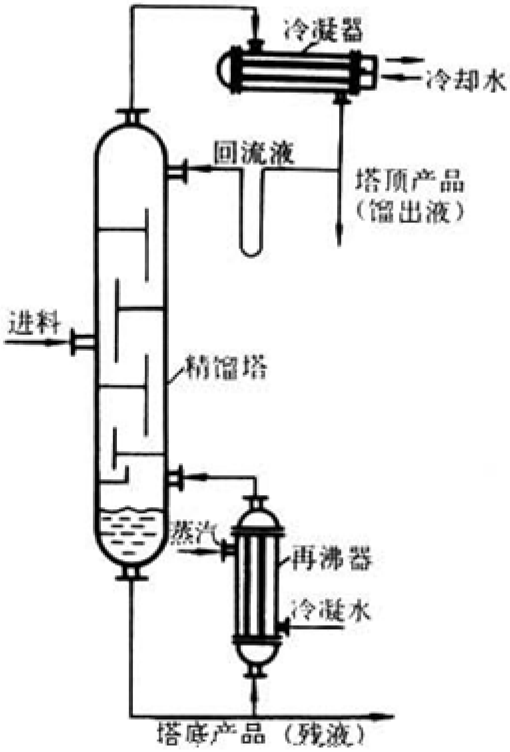
虽然我们的课程中只学习了最为经典的连续精馏部分，但却依然引申出了许多的值得讨论的话题。

首先介绍连续精馏单元操作设备，经典的板式精馏塔如下图所示：

该板式精馏塔包括，精馏塔、再沸器、冷凝器等部分。

精馏塔供汽液两相接触进行相际传质，位于塔顶的冷凝器使蒸气得到部分冷凝，部分凝液作为回流液返回塔顶，其余馏出液是塔顶产品。位于塔底的再沸器使液体部分汽化，蒸气沿塔上升，余下的液体作为塔底产品。进料加在塔的中部，进料中的液体和上塔段来的液体一起沿塔下降，进料中的蒸气和下塔段来的蒸气一起沿塔上升。在整个精馏塔中，汽液两相逆流接触，进行相际传质。液相中的易挥发组分进入汽相，汽相中的难挥发组分转入液相。对不形成恒沸物的物系，只要设计和操作得当，馏出液将是高纯度的易挥发组分，塔底产物将是高纯度的难挥发组分。进料口以上的塔段，把上升蒸气中易挥发组分进一步提浓，称为精馏段；进料口以下的塔段，从下降液体中提取易挥发组分，称为提馏段。

两段操作的结合，使液体混合物中的两个组分较完全地分离，生产出所需纯度的两种产品。



在笔者看来，精馏之所以能使液体混合物得到较完全的分离，最巧妙的地方在于“回流”。广义的“回流”包括两部分，塔顶高浓度易挥发组分液体和塔底高浓度难挥发组分蒸气两者返回塔中。汽液回流形成了逆流接触的汽液两相，从而在塔的两端分别得到相当纯净的单组分产品。因此，正如理论分析一样，实际的单元操作中，回流也是一个重要的精馏操作控制参数，它的变化影响精馏操作的分离效果和能耗。实际中只要调节回流和流出液的比例就可以改变精馏塔的工况，但同时要注意的是要保证精馏它操作在平衡稳定状态，即满足：

进料 F = 塔底出料 W + 塔顶馏出液 D

本文中只讨论精馏塔稳定后的生产问题。这个问题来源于实际应用，比如在石油化工行业中，各种成品油的价格每日都在变化，特别地，如果取一个月为观察周期，那么油价的波动很大（见下表 1），这种波动往往受到政治因素、金融因素的影响；同时原油的价格波动更加剧烈（往往见诸于新闻）。因此一个石油化工企业必须根据市场的实际行情来优化生产控制，否则企业将很难生存。

表格 1 2009 年 11 月 10 日油价调整

项目	90#汽油			93#汽油			97#汽油			0#柴油		
内容	调整前	调整后	涨幅	调整前	调整后	涨幅	调整前	调整后	涨幅	调整前	调整后	涨幅
浙江	5.50	5.85	↑0.35	5.89	6.27	↑0.38	6.26	6.66	↑0.40	5.72	6.13	↑0.41

而各种汽油实际上是精馏塔中不同塔段的侧线产品，因此本文提出的话题就是：

怎样统筹规划一个理论精馏塔的多股进料以及多股侧线出料的方案，使得全塔的经济效益最高。

由于石油化工精馏模型复杂，且笔者没有实际经验，故本文中换用“酒精——水”二元物系[1]，来做数值计算对象，该物系在实验课中有过接触，有物系的数据可供本文使用。事实上本文的内容可以推广到任意的二元物系。

建模分析

精馏塔经济因素包括产品创造经济效益，以及各种操作费用。

产品的经济效益

市场对于不同浓度的酒精会有不同浓度的酒精会有不同的报价，以提供两种产品的精馏塔为例，一种塔顶产品和一种塔侧出料产品，其市场报价分别为 p,p1。

	塔顶产品	塔侧产品	塔底产品[2]
浓度	x_D	x_{D1}	x_W
摩尔流率	D	D_1	W
售价	p	p_1	p_W

操作费用

这是一个精馏塔主要的开销度量，简单建模考虑包括原料费，包括再沸器的加热费用、冷凝器的冷却费用和精馏设备的折旧费。当操作时变动回流比，直接影响前两项费用。

	原料 1	原料 2
摩尔流率	F1	F2
进塔热状况	q_1	q_2
加热到所需热状况的单位热状况费用	*h*	
单价	c_1	c_2
塔顶冷凝摩尔流率	V	
塔顶气液相温度差	t_D	
单位塔顶冷凝费用	cold	
塔底加热摩尔流率	W	

1 物系的相对挥发度数据附后
2 在酒精——水精馏中，塔底虽然产品几乎是水，但是由于出塔的水热值高，依然有经济效益

塔底气液相温度差	t_W
单位塔底加热费用	hot

折旧费几乎不随操作变化而变化，故在模型中略去。

综合以上的精馏塔经济价值分析，可以归纳出优化模型的目标函数：

$$In = D \cdot p + D_1 \cdot p_1 + W \cdot p_W$$
$$Out = -h \cdot (F_1 \cdot q_1 + F_2 \cdot q_2) - c_1 F_1 - c_2 F_2 - cold \cdot V \cdot t_D - hot \cdot V' \cdot t_W$$
$$f = In + Out \quad (1)$$

精馏塔塔约束

优化精馏塔的操作必然是一个有约束的规划问题，塔受到塔的物理现实的约束，本文中从理想的情形考虑精馏塔的操作状况，得到以下约束：

守恒约束：

$$\mathrm{F_1 + F_2 = D_1 + D_2 + W} \quad (2)$$

操作线方程、提流线方程、加料线方程 (3)

热守恒假设（由于恒摩尔假设舍去） (4)

实际约束：

$$\text{精馏要求的塔板数}\mathrm{N_r} \le \text{实际塔板数}\mathrm{N_p} \quad (5)$$

受实际管路约束：

$$\mathrm{F_1 + F_2 \le F_{max}} \quad (6)$$

出塔产品有浓度范围约束：

如果在该约束范围内，厂家可以通过勾兑的方式修正，达到标准的市场要求的浓度。但是如果超出这个范围，就需要大量的而外成本。

$$\left|\mathrm{x_D - x_D^0}\right| \le \epsilon$$
$$\left|\mathrm{x_{D1} - x_{D1}^0}\right| \le \epsilon_1 \quad (7)$$

三个假设：

在酒精物系中有恒摩尔流假设（舍去了方程(4)）

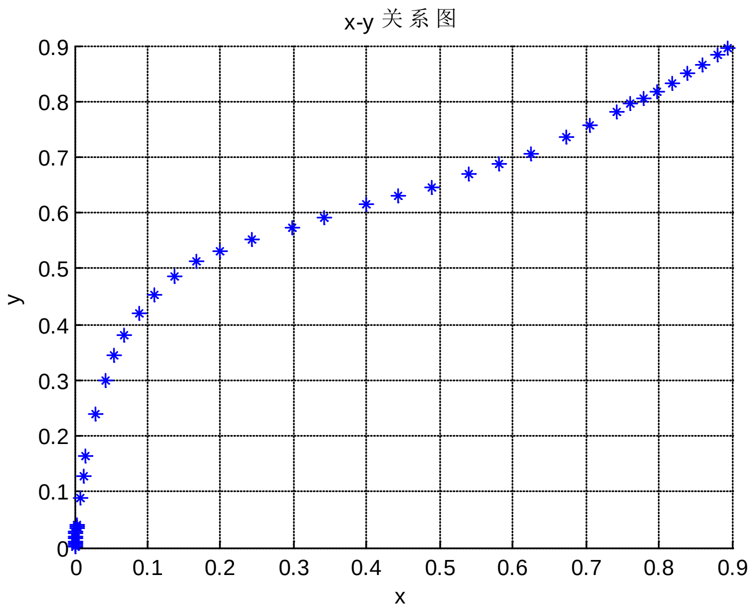
理论板假设（即假设传质充分）下的板平衡方程 (8)

泡点回流假设（此假设为了简化计算，如果不是泡点回流，要修正回流比）

数据拟合

由于精馏计算中要频繁用到平衡方程，故先对平衡方程做数值分析。

化工实验室提供的“酒精——水”物系数据作”x-y”图：



由于不是理想物系，不能用相对挥发度的方式来表示 x-y 的关系，由于每一个数据点都是精确的实验获得的，故而考虑使用线性差值的方法来进行，可以使用 M-T 画梯级方法，计算一组全回流理论塔板的操作方法，下面就是使用 matlab 脚本语言用基本循环控制流画出梯级。

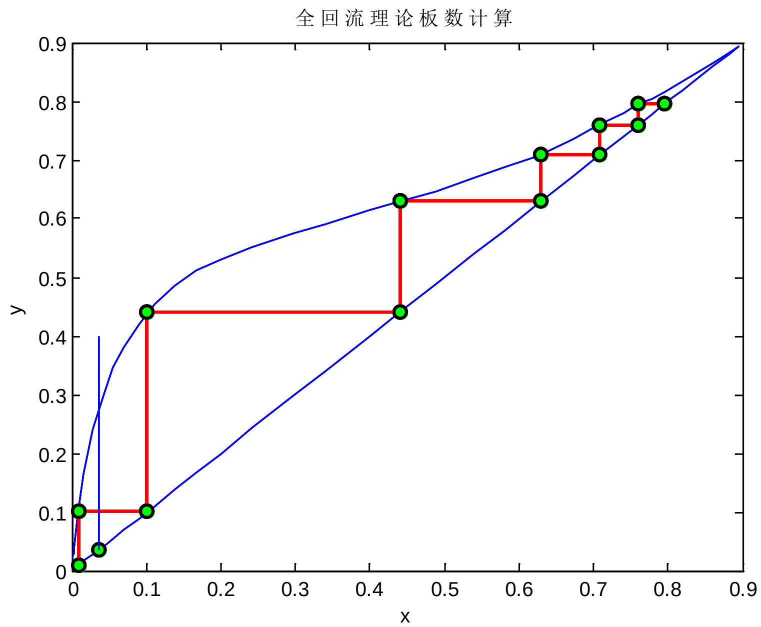
```
% draw the operation plot
plot([xD,xD-0.8],[xD,R/(R+1)*(xD-0.8)+xD/(R+1)]);
plot([xF,xF+0.2],[xF,q/(q-1)*(xF+0.2)-xF/(q-1)]);
text(xF+0.2,q/(q-1)*(xF+0.2)-xF/(q-1),'进料q线',...
    'VerticalAlignment','Top');
% draw the cross point of the operation line and the extraction line
[xd,yd] = Line_cross(R/(R+1),xD/(R+1),q/(q-1),-xF/(q-1));
     % get the cross point of 进料线 & 精馏线
plot(xd,yd,'*');
plot([xW,xd],[xW,yd],'-');   % 提馏线

axis([0,0.9,0,0.9]);
xp=xD;  % previous x(k), used in drawing picture
x(1)=inf;
n=1;
while xp>xW
   [Y,i]=max(ye>y(n));
   x(n)=(xe(i-1)-xe(i))/(ye(i-1)-ye(i))*(y(n)-ye(i))+xe(i);
   plot([xp,x(n)],[y(n),y(n)],'-ro'...
     ,'LineWidth',2,'MarkerEdgeColor','k',...
        'MarkerFaceColor','g','MarkerSize',6);
   n=n+1;
   if (x(n-1)>xd)
   y(n)=R/(R+1)*x(n-1)+xD/(R+1);    % 操作线
              % theoretical operation line and extracting line
```

```
    else  y(n)=(yd-xW)/(xd-xW) * (x(n-1)-xW) + xW;
    end
    plot([x(n-1),x(n-1)],[y(n),y(n-1)],'-ro'...
      ,'LineWidth',2,'MarkerEdgeColor','k',...
         'MarkerFaceColor','g','MarkerSize',6);
    xp=x(n-1);
end
n=n-1;
plot([xW,xW],[xW,0.15])
NT=n-1+(xW-x(n-1))/(x(n)-x(n-1))
```

Line_cross()函数用于计算直线交点，如下：

```
function [x,y]=Line_cross(a,b,c,d)
y = (a*d-c*b)/(a-c);
x = (y-b)/a;
```

下面是上程序生成的全回流情况下的理论板书计算所用的 M-T 梯级图



分析：

可以看到用分段拟合的方式来接近 y-x 关系曲线（由于数据点较多，采用线性拟合，且其误差也不大）虽然编程简单，特别是处理梯级与平衡线的交点的时候，计算直线的交点几乎没有什么计算量，只是一个简单的控制流程序。但这种计算方法的缺点也十分明显，正因为平衡线方程分段的特点，不能表示成一个整体的函数，使得我们不能解析地表述该问题。故而笔者专用多项式拟合的方法来确定该平衡关系式的拟合函数。

使用比这大作业中使用过的高阶多项式拟合方法，采用下列函数，在最小二乘的意义下逼近平衡线。

$$y = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

为了使用笔者编写过的线性规划，及其系数的置信区间[3]，可以将以上一元多项式回归

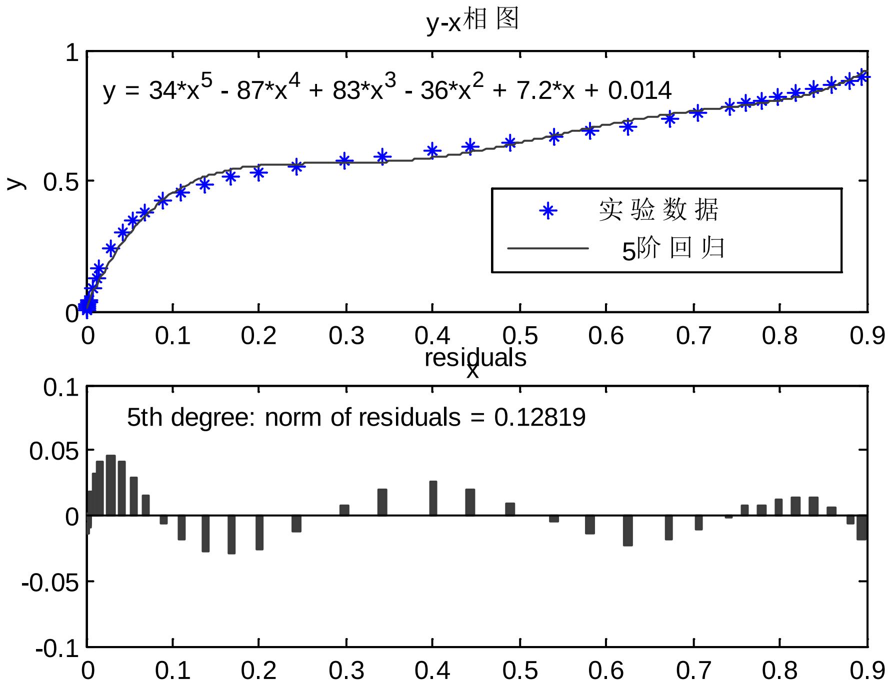
[3] 理论推导具体见笔者《差值与拟合部分作业》

转化成多元线性回归，只要令：

$$s_i = x^i$$
$$y = a_0 s_n + a_1 s_{n-1} + \cdots + a_{n-1} s + a_n$$

```
% k_order regression
X = ones(n,1);
for i=1:k
X = [X , x'.^i];  % k=1
end
A2 = X'*X;
b2 = X'*y';
a = A2\b2;
a=rot90(rot90(a));  % let the polynominal arranged in decreasing powers.
y_es = polyval(a,x);

alpha = 0.05;  % 1-alpha是置信度
C = inv(A2);
delta = tinv(1-alpha/2 , n-k-1) *...   % 系数置信区间长度
    sqrt(diag(C).*sum((y_es-y).^2)/(n-k-1));
delta=rot90(rot90(delta));     % in increasing power series
```

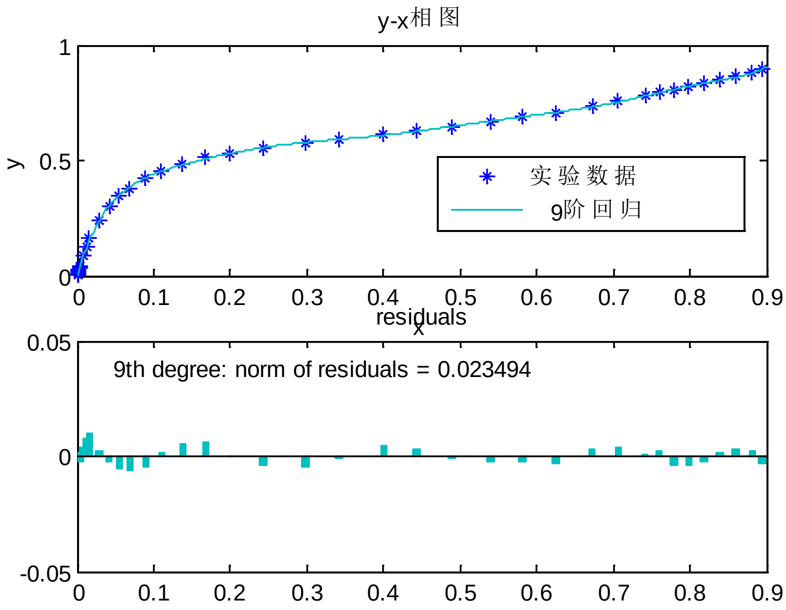
虽然我们进行多项式拟合的一半次数最高也只达到五次方，但是通过残差分析可以发现，残差呈现有规律的波动，说明残差与 x 还存在着较强的相关关系，可以认为这样的回归并不可用，于是采用更高的阶次的回归方程（如下图所示）。



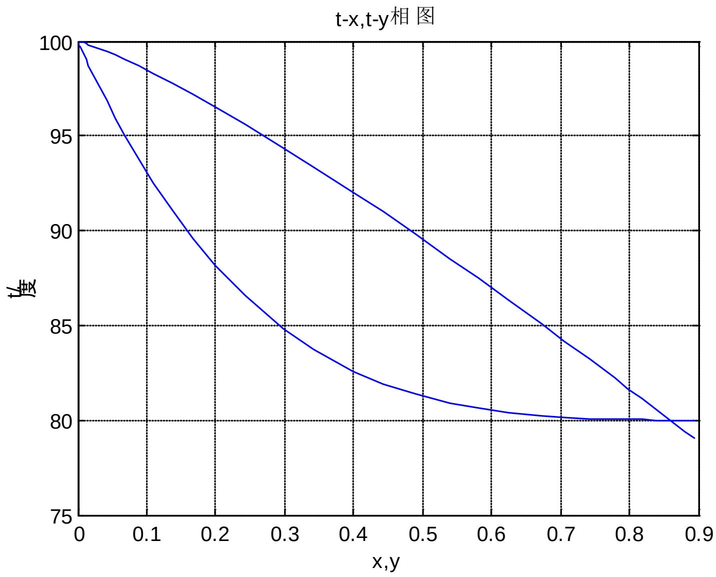
发现通过 9 次多项式回归，在我们关心的(0,1)段，y-x 关系曲线可以得到如下的回归结果：

系数	值	置信区间
a0	2238	(1613, 2863)
a1	-9565	(-1.206e+004, -7069)
a2	1.732e+004	(1.316e+004, 2.148e+004)
a3	-1.732e+004	(-2.107e+004, -1.357e+004)
a4	1.046e+004	(8483, 1.244e+004)
a5	-3929	(-4544, -3314)
a6	916.7	(808.3, 1025)
a7	-130.3	(-140, -120.6)
a8	11.29	(10.95, 11.63)
a9	0.002226	(0.0003911, 0.004061)

回归的各项统计量变现十分良好，首先可以看到各个系数的置信区间都不包含 0 点，并且区间长度与估计值的比在可以接受的范围之内（10%左右），说明回归系数真实可行。



后面将采用类似的方法求解问题。下面是用类似的方法做出了 t-x,t-y 相图。



需要说明的是，上图并不是完全的拟合结果，而是更具，教科书上的图粗糙拟合而成的。有了以上拟合出的相图多项式表示形式，就可以用解析的形式描述本模型了。

建立优化模型

显然精馏塔操作中的控制变量包括

塔顶回流比 R	侧线出料量 D1	进料口 1：F1	及其热状 q1	进料口 2：F2
及其热状况 q2	塔底出料 W			

市场决定参数（具体说明见前文）

价格：$p, p_1, p_W, h, c_1, c_2, cold, hot$,$\mathrm{x_D^0}, \mathrm{x_{D1}^0}, \epsilon, \epsilon_1$,(由于热状况 q 的物理意义，h 一般为负)
塔参数：$\mathrm{F_{max}}$,$\mathrm{N_T}$

其他参数皆为中间变量。

用以上参数表征出精馏塔的经济效益为：

$$\max\{f\}$$

$$f = (F_1 + F_2 - W)p + D_1 p_1 + W p_W - h(F_1 q_1 + F_2 q_2) - c_1 F_1 - c_2 F_2 \ldots$$

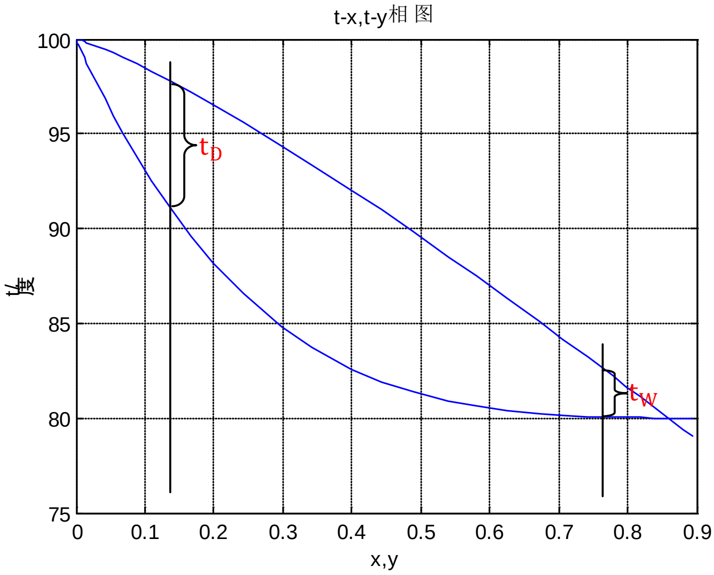
$$\ldots - cold(F_1 + F_2 - W)(1 + R)t_D - hot[F_1 q_1 + F_2 q_2 + (F_1 + F_2 - W)\mathrm{R} - \mathrm{W}]t_W$$

以上式子中所有变量、参数都是前面模型中所提供的，除了$\mathrm{t_D}$和$\mathrm{t_W}$ 以外。
根据定义，塔顶冷凝温差为：

$$\mathrm{t_D} = \mathrm{t(y_D)} - \mathrm{t(x_D)}$$

$$\mathrm{t_W} = \mathrm{t(y_W)} - \mathrm{t(x_W)}$$

在“t-x ,t-y”相图中表示出来。

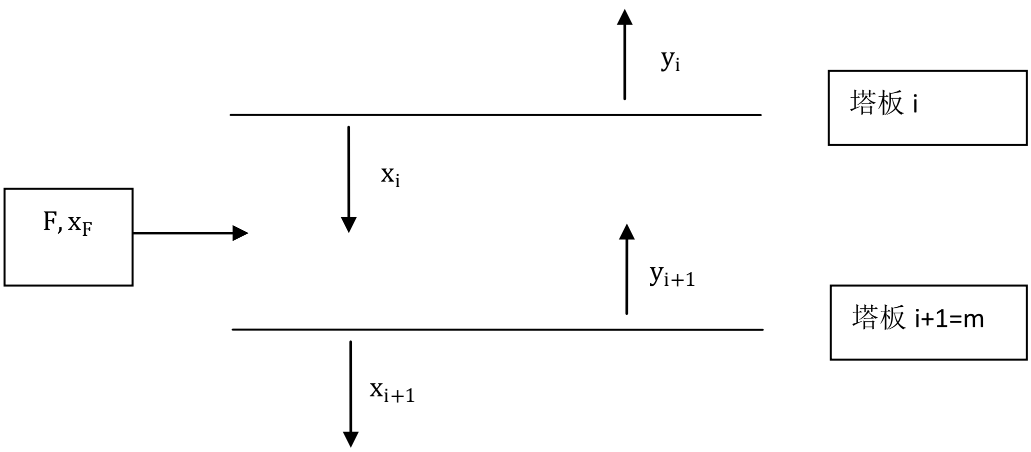


其中$\mathrm{x_D^0} = 0.80$ 可以事先确定，而$\mathrm{y_W} = \mathrm{x_W}$需要通过联立(2),(3),(4),(8)才能得出。
约束条件变量化

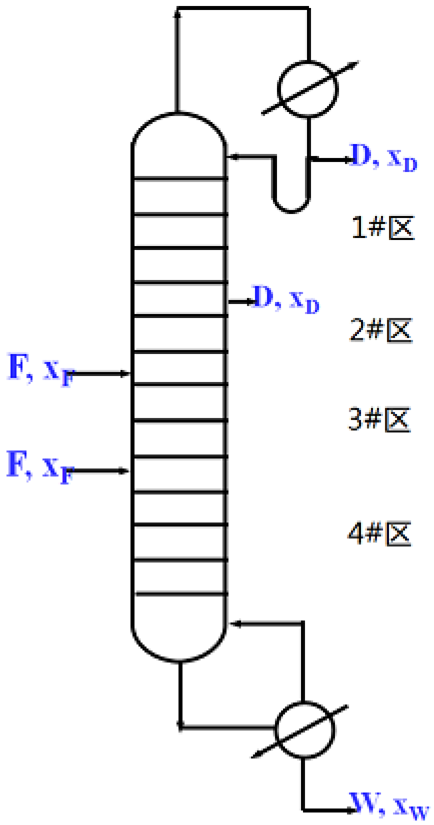
传统课本中所述的多为精馏塔理论塔板设计题，即给定塔顶要求$\mathrm{x_D}$，用 M-T 方法画出梯级，然后根据热状况，选择最佳进料点，然后设计塔高度等等。但是优化问题是一个逆问

题，即实现知道塔构造（本文中认为是筛板塔），进料板位置等等都已经确定。在确定的塔结构参数上，进行计算优化时本题的目标函数。

下面列出相邻的两块塔板的模型图：



由于前面假设一共有两个进料口，两个侧线出料口，那么按照其在精馏塔上的位置，将全塔分为 1#、2#、3#、4#共 4 个子段，在每段上有不同的物料守恒方程。



1#：

$$\begin{cases} V = D + L \\ Vy_{n+1} = Lx_n + Dx_D \end{cases}$$

2#：

$$\begin{cases} V_2 = D + D_1 + L_2 \\ V_2y_{n+1} = L_2x_n + Dx_D + D_1x_{D1} \\ L_2 = L - D\,;\ V = V_2 \end{cases}$$

3#：

$$\begin{cases} V_3 = D + D_1 + L_3 \\ V_3y_{n+1} = L_3x_n + Dx_D + D_1x_{D1} \\ V_2 = V_3 + (1 - q_1)F_1 \\ L_3 = L_2 + q_1F_1 \end{cases}$$

4#：

$$\begin{cases} V_4 = D + D_1 + L_4 \\ V_4y_{n+1} = L_4x_n + Dx_D + D_1x_{D1} \\ V_3 = V_4 + (1 - q_2)F_2 \\ L_4 = L_3 + q_2F_2 \end{cases}$$

以上便是(2)~(4)中所述的守恒约束方程，其中 V 表示上升蒸汽、L 是下降馏出液。同时 $y_n = y(x_n)$。

约束优化问题计算方法

由于本问题是一个非线性的复杂优化问题（优化目标和约束方程都是非线性的），因此可以使用拉格朗日乘子法的思想，将约束转化成目标，从而将有约束优化转化成为

无约束优化问题，然后调用经典的算法求解该问题。

为了便于进行数值计算，取下面一个精馏塔的实例进行计算：

```
% visual market parameters
p = 2;   p_1 = 1;     p_W = 0.1;    h = -1;  c_1 = 1.5;   c_2 = 0.5;
cold = 1;   hot =1;     epsilon = 1e-2;      epsilon1 = 1e-2;
xD0 = 0.85;
% rectification tower parameters
Np = 8;    %实际塔板总数
k = 4;      %侧线出料口
m = 6;      %第一进料口
n = 7;      %第二进料口
y = ones(10,1); %10块塔板的初始化
x = ones(10,1); %
```

带约束优化问题的转化

正如前面所述，将带约束的方程转化为无约束的方程，可以采用惩罚函数的方法，惩罚函数的构造从理论上说有各种方法。此处采用外点法。

不是一般性地将本文的优化问题，表述成如下形式：

$$\min f(\bar{x}), \quad \mathrm{s.t.}\begin{cases} h_i(\bar{x}) = 0 \\ g_j(\bar{x}) \geq 0 \end{cases}$$

内点法为：

$$F(\bar{x}) = f(\bar{x}) + M\sum_{i=1}^{k} h_i^2(\bar{x}) + M\sum_{j=1}^{m}\left[\min\left(0, g_j(\bar{x})\right)\right]^2$$

用上公式的构造新函数方法编写下面 MATLAB 中的 m 函数用以调用其他无约束的优化方法进行优化，具体方法在后面介绍。（注意）

```
function [x,minf] = minPF(f,x0,g,h,c1,p,var,eps)
% f - objective function
% x0 - initial point
% h – h equation
% g – g nonequation
% c1 - initial PENALTY constant
% p - proportional constant
% var - variables
% eps - precision
% minf - objective minimum value
% x - the varible when minimum stands
format long;
k = 0;
```

```
FE = 0;
for i=1:length(h)
    FE = FE + (h(i))^2;
end
x1 = transpose(x0);
x2 = inf;
while 1
    M = c1*p;
    FF = M*FE;
    gx = Funval(g,var,x1);
    gF = 0;
    for i = 1:length(h)
        if gx(i) < 0
            gF = gF + M*(g(x)^2);       % penalty function
        end
    end
    SumF = f + FF + gF;

    [x2,minf] = minMethod (SumF,transpose(x1),var);
                                        % non-Constraint Algorithm
    if norm(x2 - x1)<=eps
        x = x2;
        break;
    else
        c1 = M;
        x1 = x2;
    end
end
minf = Funval(f,var,x);
format short;
```

下面按照前面的既定规则，将约束方程表示出来，其中 *h* 表示等式约束，*g* 表示不等式约束，不等号方向如(9)式约定。

所有变量用符号变量，易于编程实现，ff1~ff10 表示塔板上的平衡浓度，hh0~hh2 表示各种守恒方程的约束形式。

```
syms R D_1 F_1 q_1 F_2 q_2 W t_D t_W D V L V_2 L_2 V_3 L_3 L_4 V_4;
syms x1 x2 x3 x4 x5 x6 x7 x8;
syms y1 y2 y3 y4 y5 y6 y7 y8;
p = [2238   -9565   17320   -17320  10460   -3929 ...
    916.7   -130.3  11.29   0.002226];
pp = [-20 80 -120 -24 80];
f=(F_1+F_2-W)*p+D_1* p_1+W*p_W-h*(F_1* q_1+F_2 *q_2 )-c_1* F_1-c_2*...
    F_2 - cold*(F_1+F_2-W)*(1+R) * t_D-hot*(F_1* q_1+F_2* ...
    q_2+(F_1+F_2-W)*R-W)* t_W;
ff1 = p * [x1^9; x1^8; x1^7; x1^6; x1^5; x1^4; x1^3; x1^2; x1; 1] -y1;
```

```
ff2 = p * [x2^9; x2^8; x2^7; x2^6; x2^5; x2^4; x2^3; x2^2; x2; 1] -y2;
ff3 = p * [x3^9; x3^8; x3^7; x3^6; x3^5; x3^4; x3^3; x3^2; x3; 1] -y3;
ff4 = p * [x4^9; x4^8; x4^7; x4^6; x4^5; x4^4; x4^3; x4^2; x4; 1] -y4;
ff5 = p * [x5^9; x5^8; x5^7; x5^6; x5^5; x5^4; x5^3; x5^2; x5; 1] -y5;
ff6 = p * [x6^9; x6^8; x6^7; x6^6; x6^5; x6^4; x6^3; x6^2 ;x6; 1] -y6;
ff7 = p * [x7^9; x7^8; x7^7; x7^6; x7^5; x7^4; x7^3; x7^2; x7; 1] -y7;
ff8 = p * [x8^9; x8^8; x8^7; x8^6; x8^5; x8^4; x8^3; x8^2; x8; 1] -y8;

hh0 = [ F_1 + F_2 - D_1 - D_2 - W;
       V-D-L;
       V_2-D-D_1-L_2;
       V_3-D-D_1-L_3;
       V_4-D-D_1-L_4 ];
hh1 = [ V*y2 - L*x1 - D*y1;
       V*y3 - L*x2 - D*y1;
       V_2*y4 - L_2*x3 - D*y1 - D_1*x2;
       V_2*y5 - L_2*x4 - D*y1 - D_1*x2;
       V_3*y6 - L_3*x5 - D*y1 - D_1*x2;
       V_3*y7 - L_3*x6 - D*y1 - D_1*x2;
       V_4*y8 - L_4*x7 - D*y1 - D_1*x2;];
hh2 = [V_2 - V;
      L_2 - L + D;
      V_2 - V_3 - (1-q_1)*F_1;
      L_3 - L_2 - q_1*F_1;
      V_3 - V_4 - (1-q_2)*F_2;
      L_4 - L_3 - q_2*F_2];

T = [t_D - pp*[y1^4; y1^3; y1^2; y1^1.2; y1];
    t_D - pp*[y1^4; y1^3; y1^2; y1^1.2; y1]];
h = [ ff1; ff2; ff3; ff4; ff5; ff6; ff7; ff8; hh0; hh1; hh2; T];
g = [   Fmax - F_1 - F_2;
       epsilon + xD0 - y1 ];
```

通过以上程序，一个较为复杂的带约束优化问题就转化为了有约束的优化问题。

无约束优化问题的解法

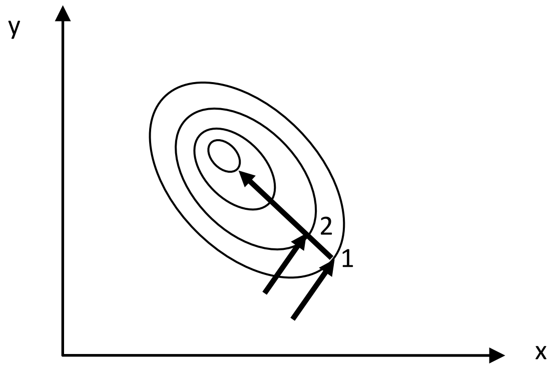
通过自学无约束优化问题，一类基于函数性态（只要是可微性质）的方法，发展 出了牛顿梯度方法以及最速上升方法等。但是这种方法对函数的自变量比较敏感，最麻烦之处在于这些方法要求计算待优化函数的全微分[[4]]。可以看到本问题中的变量较多，各个变量的耦合程度高，最终的独立变量也很多，故而综合考虑使用直接方法求解，直接

[4] 见《工程数值方法》第 5 版，378 页

方法中Powell方法比较出色。

《工程数值方法》中表述了该方法的思路：

*“若点 1 和点 2 可以通过在同一方向上但是不同起点的一位检索获得的话，那么由点 1 和点 2 构成的直线将直接指向最大值。这条线被称作是共轭方向。”*如下图所示。



根据这一思想，可以得到如下的算法叙述：

1) 给定初始点$x^{(0)}$，n 个线性无关的初始向量组$\{p^0,p^1,\dots,p^n\}_0$及精度$\epsilon > 0$，置 k=0；
2) 令$y^0 = x^k$，依次沿 $\{p^0,p^1,\dots,p^n\}_k$ 中的方向进行一维搜索：

$$\begin{cases} y^j = y^{j-1} + t_{j-1}p^{j-1} \\ f\left(y^{j-1} + t_{j-1}p^{j-1}\right) = \min f\left(y^{j-1} + tp^{j-1}\right) \end{cases}, \quad j = 1,2,\dots,n;$$

3) 令$p^n = y^n - y^0$，若$\|p^n\| \le \epsilon$，停止迭代，输出$x^{k+1} = y^n$
4) 求出 m，使得

$$f(y^{m-1}) - f(y^m) = \max\{f\left(y^{j-1}\right) - f\left(y^j\right), 1 \le j \le n\}$$

若下面式子成立

$$f(y^0) - 2f(y^n) + f(2y^n - y^0) < 2[f(y^m - 1) - 2f(y^m)]$$

转 5)，否则 6)

5) 求解$\min f(y^n + tp^n)$，设最优解为t_n，令

$$\begin{cases} x^{k+1} = y^n + t_np^n \\ \{p^0,p^1,\dots,p^{n-1}\}_{k+1} = \{p^0,p^1,\dots,p^{m-1},p^{m+1},\dots,p^n\}_k \\ k = k+1 \end{cases}$$

转 2)

6) 令$x^{k+1} = y^n$，置 k=K+1，转 2)

用 MATLAB 中的 m 函数编写如下程序：

```
function [x,minf] = minMethod(f,x0,P,var,eps)
% f - objective function
% x0 - initial vector
% P - vector group
% var - variable
% eps - precision
format long;
n = length(var)+1;
syms l;

while 1
    y = zeros(size(P));
```

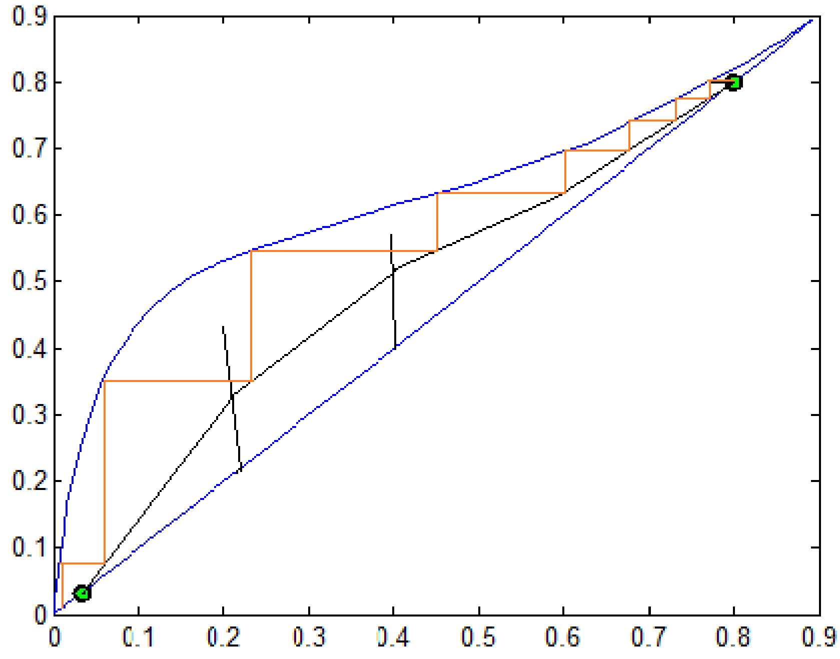
```
    y(:,1) = x0;
    for i=1:n-1
        yv = y(:,i) + l*P(:,i);
        fy = Funval(f, var,yv);
        [a,b] = Funval(fy,0,0.1);
        tl = min(a,b);
        y(:,i+1) = y(:,i) + tl*P(:,i);
    end
    P(:,n) = y(:,n) - y(:,1);
    if norm(P(:,n)) <= eps
        x = y(:,n);
        break;
    else
        for j=1:n
            FY(j) = Funval(f, var,y(:,j));
        end
        maxDF = -inf;
        m = 0;
        for j=1:n-1
            df = FY(j) - FY(j+1);
            if df > maxDF
                maxDF = df;
                m = j+1;
            end
        end
        tmpF = Funval(f, var,2*y(:,n)-y(:,1));
        fl = FY(1) - 2*FY(n) + tmpF;
        if fl<2*maxDF
            yv = y(:,n) + l*P(:,n);
            fy = Funval(f, var,yv);
            [a,b] = Funval (fy,0,0.1);
            tl = min(a,b);
            x0 = y(:,n) + tl*P(:,n);
            P(:,m:(n-1)) = P(:,(m+1):n);
        else
            x0 = y(:,n);
        end
    end
end
minf = Funval(f,var,x);
format short;
```

其中Funval函数[5]如下:

[5] 该程序可见于 CSDN 网站中

```
function fv = Funval(f,varvec,varval)
var = findsym(f);
varc = findsym(varvec);
s1 = length(var);
s2 = length(varc);
m =floor((s1-1)/3+1);
varv = zeros(1,m);

if s1 ~= s2
for i=0: ((s1-1)/3)
k = findstr(varc,var(3*i+1));
index = (k-1)/3;
varv(i+1) = varval(index+1);
end
fv = subs(f,var,varv);
else
fv = subs(f,varvec,varval);
end
```

由于笔者没有充足的时间得到实际经济参考价值的参数，假设了前文实例中的各种参数，最后得到数值结果图示如下（重要的参数 R=4.032；q_1=0.951；q_2=0.848；F1=4.652； F2=5.348； D=3.05； D1=3.57； W=3.38；出料口浓度达到要求）。



小结

本从上课的内容提出自己想法，通过查阅资料，提出了问题的严格形式，然后通过数学建模，设定了一些松弛变量、自由变量，成功的将模型表达成了标准的非线性带约束的优化形式，再次过程中还应用了以前曾经研究过的函数拟合的内容，将要运用在本问题的计算当中。

首先高度非线性的约束优化问题求解难度高，本文中采用了惩罚函数中的“内点法”将其转化成为一个无约束的优化问题。对于得到的无约束优化标准模型，由于变量多、耦合程度高、以及目标函数形式复杂，采用了 Powell 方法来计算，在计算本文的实例之前还用程序计算过了一个简单的二元函数，并与 MATLAB 的 fminsearch()函数计算了该函数的实例，得到了完全相同的答案，验证了本算法的正确性。

假以更多的时间，还将运用数值分析的手段对优化模型的稳定性以及灵敏度进行分析。