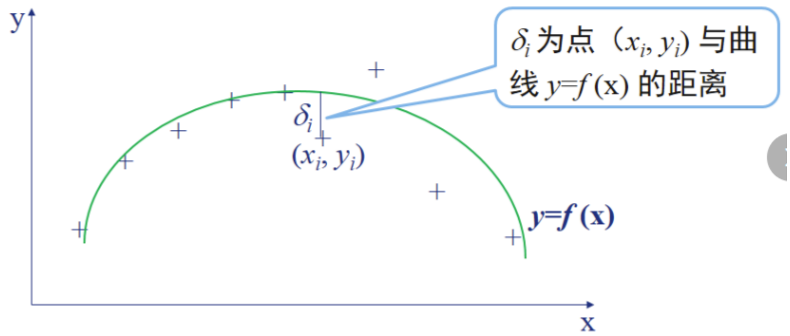
拟合

曲线拟合

- 已知一组（二维）数据，即平面上 n个点(x_i, y_i)（i=1,…n），寻求一个函数（曲线）$y=f(x)$, 使 $f(x)$ 在某种准则下与所有数据点最为接近，即曲线拟合得最好。



拟合与插值

- **问题**：给定一批数据点，需确定满足特定要求的曲线或曲面。
- **解决方案**：
  - **插值问题**：要求所求曲线（面）通过所给所有数据点。
  - **数据拟合**：不要求曲线（面）通过所有数据点，而是要求它反映对象整体的变化趋势。
  - 函数插值与曲线拟合都是要根据一组数据构造一个函数作为近似，由于近似的要求不同，二者的数学方法上是完全不同的。

曲线拟合常用解法 — 线性最小二乘法

- 选定一组函数 $r_1(x), r_2(x), \ldots r_m(x)$, $m<n$,

  令 $f(x)=a_1r_1(x)+a_2r_2(x)+\ldots+a_mr_m(x)$

  其中 $a_1, a_2, \ldots, a_m$ 为待定系数。
- 确定$a_1, a_2, \ldots, a_m$ 的准则（最小二乘准则）：

  使n个点（x_i, y_i）与曲线 $y=f(x)$ 的距离δ_i的平方和最小 。

$$J(a_1, a_2, \cdots a_m) = \sum_{i=1}^{n} \delta_i^2 = \sum_{i=1}^{n}[f(x_i) - y_i]^2 = \sum_{i=1}^{n}[\sum_{k=1}^{m} a_k r_k(x_i) - y_i]^2$$

求 $a_1, a_2, \ldots, a_m$
使 $J(a_1, a_2, \ldots, a_m)$ 最小

线性最小二乘法的求解—预备知识

➢ 超定方程组：方程个数大于未知量个数的方程组

$$\begin{cases} r_{11}a_1 + r_{12}a_2 + \cdots + r_{1m}a_m = y_1 \\ \cdots\cdots \\ r_{n1}a_1 + r_{n2}a_2 + \cdots + r_{nm}a_m = y_n \end{cases} \quad (n > m) \qquad \text{即}\ \boldsymbol{Ra} = \boldsymbol{y}$$

其中 $R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}, \quad a = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$

➢ 超定方程一般是不存在解的矛盾方程组。

➢ 如果有向量 $\boldsymbol{a}$ 使得 $\sum_{i=1}^{n}(r_{i1}a_1 + r_{i2}a_2 + \cdots + r_{im}a_m - y_i)^2$ 达到最小，则称 $\boldsymbol{a}$ 为上述超定方程的最小二乘解。

□ 曲线拟合的最小二乘法要解决的问题，

实际上就是求以下超定方程组的最小二乘解的问题。

$\boldsymbol{Ra} = \boldsymbol{y}$

其中 $R = \begin{bmatrix} r_1(x_1) & \cdots & r_m(x_1) \\ \cdots\cdots & & \\ r_1(x_n) & \cdots & r_m(x_n) \end{bmatrix}, \quad a = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$

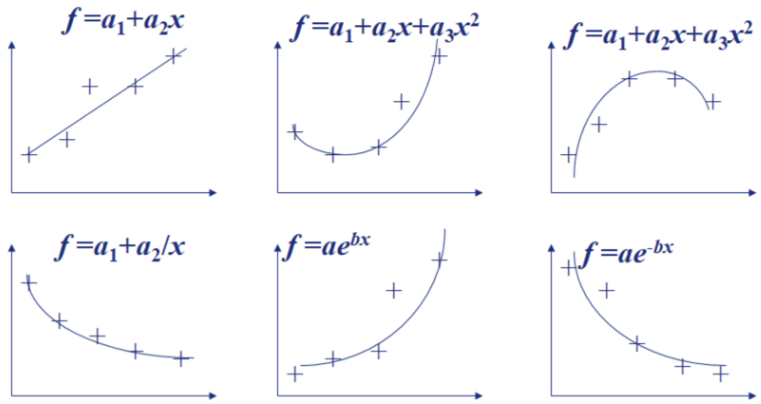
□ 定理：当R^TR可逆时，超定方程组存在最小二乘解，且即为方程组

$$R^TRa = R^Ty$$

的解：$\boldsymbol{a} = (\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{R}^T\boldsymbol{y}$

➢ 通过机理分析建立数学模型来确定 $f(x)$；

➢ 将数据 (x_i, y_i) $(i=1, \ldots, n)$作图，通过直观判断确定 $f(x)$：



- 多项式拟合： $f(x)=a_1x^m+\ldots+a_mx+a_{m+1}$
- 函数命令polyfit

a = polyfit(x, y, m)

输出拟合多项式系数 $a=[a_1,\ldots a_m, a_{m+1}]$

输入同长度的数组X, Y

拟合多项式次数

- 多项式在 x 处的值 y 的计算： y = polyval(a, x)
- 超定方程组 $R_{n\times m}a_{m\times 1}=y_{n\times 1}(m<n)$用 a=R\y (MATLAB中)可得最小二乘意义下的解。

- **例** 对下面一组数据作二次多项式拟合

xi	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
yi	-0.447	1.978	3.28	6.16	7.08	7.34	7.66	9.58	9.48	9.30	11.2

- **分析**：即要求出二次多项式 $f(x)=a_1x^2+a_2x+a_3$ 中的 $A=(a_1,a_2,a_3)$， 使得：

$$\sum_{i=1}^{11}[f(x_i)-y_i]^2 \quad \text{最小。}$$

解法1：用解超定方程的方法

$$R=\begin{pmatrix} x_1^2 & x_1 & 1 \\ \cdots & \cdots & \cdots \\ x_{11}^2 & x_{11} & 1 \end{pmatrix}$$

- 输入以下命令：

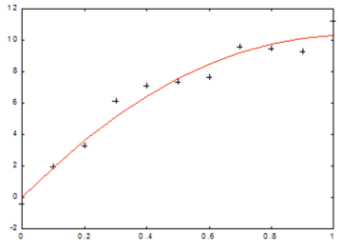
```
x=0:0.1:1;
y=[-0.447 1.978 3.28 6.16 7.08 7.34 7.66 9.56 9.48 9.30 11.2]
R=[(x.^2)' x' ones(11,1)];
A=R\y'
```

- 计算结果 A = -9.8108 20.1293 -0.0317
- $f(x)=-9.8108x^2+20.1293x-0.0317$

➢ 解法2：用多项式拟合的命令

● 输入以下命令：

```
x=0:0.1:1;
y=[-0.447 1.978 3.28 6.16 7.08 7.34
A=polyfit(x,y,2)
z=polyval(A,x);
plot(x,y,'k+',x,z,'r')      %作出数据点和拟合曲线的图形
```



● 计算结果 A ＝-9.8108   20.1293   -0.0317

● $f(x) = -9.8108x^2 + 20.1293x - 0.0317$

□ Matlab提供两个求非线性最小二乘拟合的函数：

➢ **lsqcurvefit**

➢ **lsqnonlin**

➢ 两个命令都要先建立M-文件fun.m，在其中定义函数*f (x)*，但两者定义*f (x)* 的方式是不同的。

➢ 已知数据点：

$$xdata = (xdata_1, xdata_2, \ldots, xdata_n)$$

$$ydata = (ydata_1, ydata_2, \ldots, ydata_n)$$

➢ 含参量 *x*（向量）的向量值函数

$$F(x, xdata) = (F(x, xdata_1), \ldots, F(x, xdata_n))^T$$

➢ 求 *x* 使得 $\sum_{i=1}^{n}(F(x, xdata_i) - ydata_i)^2$ 最小。

非线性最小二乘拟合— lsqcurvefit

x = lsqcurvefit (‘fun’,x0,xdata,ydata,options)

fun是定义函数$F(x,xdata)$ 的m-文件, 自变量为x和xdata

迭代初值

已知数据点

选项见无约束优化

➢输入：

(1) x=lsqcurvefit (‘fun’,x0,xdata,ydata);

(2) x=lsqcurvefit (‘fun’,x0,xdata,ydata,options);

(3) [x, options]=lsqcurvefit (‘fun’,x0,xdata,ydata,…);

……

➢输出目标函数值： f=fun(x,xdata)

➢ 已知数据点：

$$xdata = (xdata_1 , xdata_2 , \dots , xdata_n)$$

$$ydata = (ydata_1 , ydata_2 , \dots , ydata_n)$$

➢ 含参量 x（向量）的向量值函数

$f(x)=(f_1(x), f_2(x), \dots, f_n(x))^T$

其中 $f_i(x) = f(x, xdata_i, ydata_i)=F(x, xdata_i)-ydata_i$

➢ 求 x 使得 $f^T(x)f(x) = f_1(x)^2 + f_2(x)^2 + \cdots + f_n(x)^2$ 最小。

x= lsqnonlin (‘fun’， x0， options)

fun是定义函数 $f(x)$ 的M-文件 自变量为 x

迭代初值

选项见无约束优化

➢ 输入：

x=**lsqnonlin**（‘fun’，x0）；

x=lsqnonlin （‘fun’，x0，options）；

x=lsqnonlin （‘fun’，x0，options， ‘grad’）；

[x，options]= lsqnonlin （‘fun’，x0，…）；

➢ 例：用下面一组数据拟合

$c(t)=a+be^{0.02kt}$ 中的参数 a, b, k 。

t_j	100	200	300	400	500	600	700	800	900	1000
$C_j/10^{-3}$	4.54	4.99	5.35	5.65	5.90	6.10	6.26	6.39	6.50	6.59

➢ 分析：该问题即解最优化问题

$$\min F(a,b,k) = \sum_{j=1}^{10}[a + be^{-0.02kt_j} - c_j]^2$$

➢ **解法一:** 用命令lsqcurvefit

$F(x, tdata)=(a+be^{-0.02kt1}, \dots, a+be^{-0.02kt10})^T$ $x=(a, b, k)$

● 编写M-文件 curvefun1.m

```
function f=curvefun1(x, tdata)
        f=x(1)+x(2)*exp(-0.02*x(3)*tdata)
                %其中 x(1)=a;  x(2)=b;  x(3)=k;
```

● 输入命令

```
tdata=100:100:1000
cdata=1e-03*[4.54,4.99,5.35,5.65,5.90,6.10,6.26,6.39,6.50,6.59];
 x0=[0.2,0.05,0.05];
 x=lsqcurvefit ('curvefun1',x0,tdata,cdata)
 f=curvefun1(x,tdata)
```

➢ 解法二：用命令lsqnonlin

$f(x)=F(x, tdata,cdata)=(a+be^{-0.02kt1}-c_1, \dots , a+be^{-0.02kt10}-c_{10})^T$

$x=(a, b, k)$

函数curvefun2的自变量是x，cdata和tdata是已知参数，故应将cdata, tdata的值写在curvefun2.m中

- 编写M-文件 curvefun2.m

```
function f=curvefun2(x)
tdata=100:100:1000;
cdata=1e-03*[4.54,4.99,5.35,5.65,5.90,6.10,6.26,6.39,6.50,6.59];
f=x(1)+x(2)*exp(-0.02*x(3)*tdata)- cdata
```

- 输入命令:

```
x0=[0.2,0.05,0.05];
x=lsqnonlin('curvefun2',x0)
f=curvefun2(x)
```

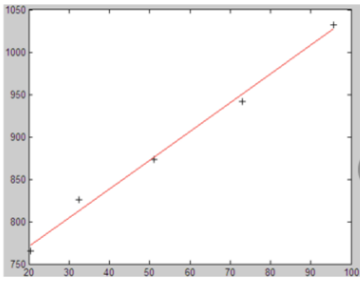
□ 例：由以下数据，拟合$R=at+b$

温度t(°C)	20.5	32.7	51.0	73.0	95.7
电阻R(Ω)	765	826	873	942	1032

➢ 方法一：用命令polyfit(x,y,m)

- 输入

```
t=[20.5 32.5 51 73 95.7];
r=[765 826 873 942 1032];
aa=polyfit(t,r,1);
a=aa(1)
b=aa(2)
y=polyval(aa,t);
plot(t,r,'k+',t,y, 'r')
```

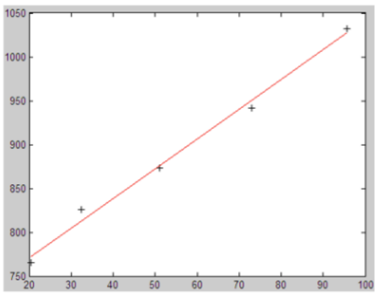


- 输出： a=3.3940, b=702.4918

➢ 方法二：直接用命令 $a = R \backslash y$

- 输入

```
t=[20.5 32.5 51 73 95.7];
r=[765 826 873 942 1032];
R=[t' ones(5,1)];
aa=R\r’;
a=aa(1)
b=aa(2)
y=polyval(aa,t);
plot(t,r,'k+',t,y,'r')
```



- 输出： a=3.3940, b=702.4918

□ 新药用于临床之前，必须设计给药方案。

➢ 血药浓度：药物进入机体后血液输送到全身，在这个过程中不断地被吸收、分布、代谢，最终排出体外，药物在血液中的浓度，即单位体积血液中的药物含量，称为血药浓度。

➢ 一室模型：将整个机体看作一个房室，称中心室，室内血药浓度是均匀的。快速静脉注射后，浓度立即上升；然后迅速下降。当浓度太低时，达不到预期的治疗效果；当浓度太高，又可能导致药物中毒或副作用太强。临床上，每种药物有一个最小有效浓度c_1和一个最大有效浓度c_2。设计给药方案时，要使血药浓度保持在c_1～c_2之间。本题设$c_1=10$，$c_2=25$(ug/ml)。

➢ 要设计给药方案,必须知道给药后血药浓度随时间变化的规律。

➢ 实验方面,，对某人用快速静脉注射方式一次注入该药物300mg后,在一定时刻t(小时)采集血液，测得血药浓度c(μg/ml)如下表:

t(h)	0.25	0.5	1	1.5	2	3	4	6	8
c(μg/ml)	19.21	18.15	15.36	14.10	12.89	9.32	7.45	5.24	3.01

➢ **问题**：

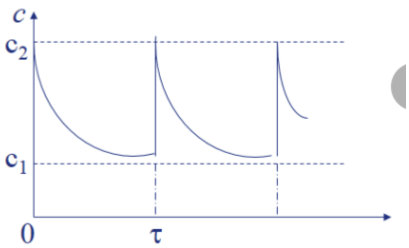
（1）在快速静脉注射的给药方式下，研究血药浓度（单位体积血液中的药物含量）的变化规律。

（2）给定药物的最小有效浓度和最大治疗浓度，设计给药方案：每次注射剂量多大；间隔时间多长。

➢ **分析**：

实验：对血药浓度数据作拟合，符合负指数变化规律。

理论：用一室模型研究血药浓度变化规律。



➢ 模型假设

（1）机体看作一个房室，室内血药浓度均匀 — 一室模型；

（2）药物排除速率与血药浓度成正比，比例系数 k(>0)；

（3）血液容积v，t=0注射剂量d，血药浓度立即为d/v。

➢ 模型建立

由假设（2）得：$\frac{dc}{dt}=-kc$

由假设（3）得：$c(0)=d/v$

所以 $c(t)=\frac{d}{v}e^{-kt}$

d=300mg，t及c(t)在某些点处的值见前表。

➢ 需经拟合求出参数k、v 。

方法一：用线性最小二乘拟合$c(t)$

$$\left.\begin{array}{l} c(t)=\frac{d}{v}e^{-kt}\Rightarrow \ln c=\ln(d/v)-kt \\ y=\ln c,\quad a_1=-k,\quad a_2=\ln(d/v) \end{array}\right\}\Rightarrow \begin{array}{c} y=a_1t+a_2 \\ k=-a_1, v=d/e^{a_2} \end{array}$$

- 输入：

```
d=300;
t=[0.25 0.5 1 1.5 2 3 4 6 8];
c=[19.21 18.15 15.36 14.10 12.89 9.32 7.45 5.24 3.01];
y=log(c);
a=polyfit(t,y,1)
k=-a(1)
v=d/exp(a(2))
```

- 输出：$k=0.2347(1/h)$， $v=15.02(l)$

方法二：用非线性最小二次拟合$c(t)$

- M文件定义函数：

```
function  f=curvefun3(x,tdata);
d=300;
f=(x(1) \d)*exp(-x(2)*tdata);
```

- 主程序：

```
t=[0.25 0.5 1 1.5 2 3 4 6 8];
c=[19.21 18.15 15.36 14.10 12.89 9.32 7.45 5.24 3.01
x0=[10,0.5];
x=lsqcurvefit('curvefun3',x0,t,c);
f=curvefun3(x,t)
x
```

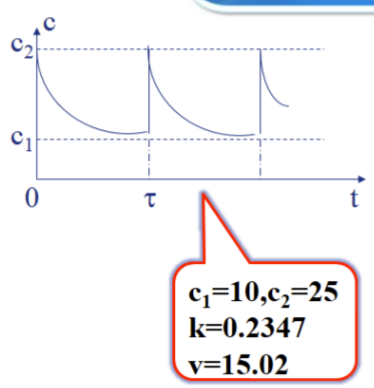
- 计算结果：$k=0.2420(1/h)$， $v=14.8212(l)$

分析：

设每次注射剂量D, 间隔时间τ

血药浓度$c(t)$ 应$c_1\le c(t)\le c_2$

初次剂量D_0应加大



- 给药方案记为：$\{D_0, D, \tau\}$

$D_0=vc_2$ ， $D=v(c_2-c_1)$

$c_1=c_2e^{-k\tau}$

所以 $\tau=\frac{1}{k}\ln\frac{c_2}{c_1}$

- 计算结果：$D_0=375.5$，$D=225.3$，$\tau=3.9$

给药方案：

$$D_0=375(mg),\quad D=225(mg),\quad \tau=4(h)$$

- 首次注射375mg，

  其余每次注射225mg，

  注射的间隔时间为4小时。