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

数据插值

数据插值 (Interpolation)

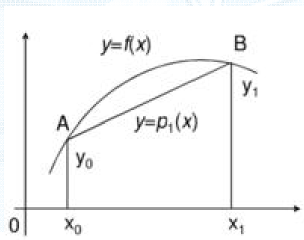
- 化工中的许多实验数据以列表函数或者表格的形式存在，如许多物性数据手册中以列表的形式给出了不同温度、压力和组成条件下的大量物性数据。
- 实际使用时，表中所列的数据往往不够，由于工艺设计的要求，需要获得介于表中两个数据之间的参数值。
- 插值：根据已知数据估算出表中未出现的数值
- 插值的数学定义：
- 已知一组离散的数据点集，在集合内部某两个点之间预测函数值的方法。用一个简单的多项式函数在某种误差范围内近似的代替原目标函数关系式。

线性插值

- 已知两个数据点$(x0,y0)$与$(x1,y1)$,要得到$[x0,x1]$区间内某一位置x上的y值。
- 构造线性插值函数：$y=a0+a1x$，用该函数计算x处的y值

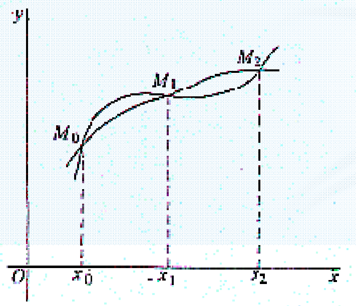
$$y\ (x)=y_0+\frac{y_1-y_0}{x_1-x_0}(x-x_0)$$

$$=\sum_{j=0}^{1}y_j\prod_{i=0}^{1}(\frac{x-x_i}{x_j-x_i})$$



二次插值(抛物线插值，拉格朗日三点插值)

- 已知三个数据点$(x0,y0)$，$(x1,y1)$和$(x2,y2)$,要得到$[x0,x2]$区间内某一位置x上的y值。
- 构造插值函数：$y=a_0+a_1x+a_2x^2$，用该函数计算x处的y值



$$\Phi\ (x)=\sum_{j=0}^{2}y_j\prod_{i=0}^{2}(\frac{x-x_i}{x_j-x_i})$$

$$(i\neq j)$$

- 最近插值法 （Nearest）
- 样条曲线插值法 （Spline）：
- 提高插值曲线光滑程度
- 埃尔米特法 （Hermite）：
- 插值函数在插值结点处函数值、导数值均与被插值函数相等的方法
- 其他插值方法

Mworks一维插值函数求解

- vq = interp1(x,v,xq) 使用线性插值返回一维函数在特定查询点的插入值。
- 向量 x 包含样本点，v 包含对应值 v(x)。向量 xq 包含查询点的坐标。如果有多个在同一点坐标采样的数据集，则可以将 v 以数组的形式进行传递。数组 v 的**每一列**都包含一组不同的一维样本值。
- vq = interp1(x,v,xq,method) 指定备选插值方法："linear"、"nearest"、"next"、"previous"、"pchip"、"cubic"、"v5cubic"、"makima" 或 "spline"。默认方法为 "linear"。
- vq = interp1(x,v,xq,method,extrapolation) 用于指定外插策略，来计算落在 x 域范围外的点。
- 如果希望使用 methodp 算法进行外插，可将 extrapolation 设置为 methodp。您也可以指定一个标量值，这种情况下，interp1 将为所有落在 x 域范围外的点返回该标量值。

例题

- 水黏度随温度的列表，求温度为25℃ 时水的黏度

T/℃	0	10	20	30	40	50	60
μ/mPa.s	1.788	1.305	1.004	0.8012	0.6532	0.5492	0.4698

- 线性插值代数计算：x=25, y的数值？

- X0=20, y0=1.004; x1=30, y1=0.8012
- y=1.004+(0.8012-1.004)*(25-20)/(30-20)
- y = 0.9026

Mworks求解

- 求温度为25℃ 时水的黏度

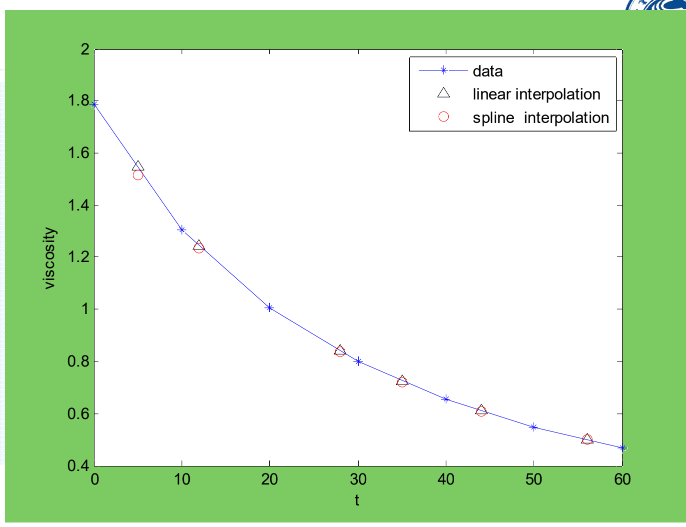
T/℃	0	10	20	30	40	50	60
μ/mPa.s	1.788	1.305	1.004	0.8012	0.6532	0.5492	0.4698

- x=[20,30];y=[1.004,0.8012]; xi=25;
- yi=interp1(x,y,xi,"linear")
- yi = 0.9026

- 求温度为5、12、28、35、44、56 ℃ 时水的黏度？
- 选用线性法和样条曲线法插值

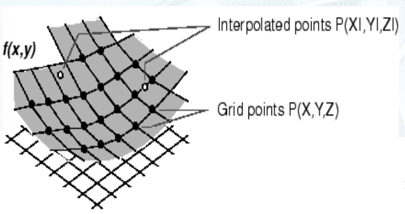
Mworks求解

- x=[0 10 20 30 40 50 60];
- y=[1.788 1.305 1.004 0.8012 0.6532 0.5492 0.4698];
- xi=[5 12 28 35 44 56];
- y_linear=interp1(x,y,xi)
- y_spline=interp1(x,y,xi,"spline")
- figure
- plot(x,y,"-*",xi,y_linear,"k^",xi,y_spline,"ro")
- xlabel ("t")
- ylabel ("viscosity")
- legend("data","linear interpolation", "spline interpolation")
- ------------------
- y_linear = 1.5465 1.2448 0.84176 0.7272 0.6116 0.5016
- y_spline = 1.51799 1.2332 0.8366 0.7212 0.6072 0.5001



Mworks二维插值函数求解

- Vq = interp2(X,Y,V,Xq,Yq) 使用线性插值返回双变量函数在特定查询点的插入值。结果始终穿过函数的原始采样。X 和 Y 包含样本点的坐标。V 包含各样本点处的对应函数值。Xq 和 Yq 包含查询点的坐标。
- Vq = interp2(V,Xq,Yq) 假定一个默认的样本点网格。默认网格点覆盖矩形区域 X=1:n 和 Y=1:m，其中 m,n = size(V)。如果您希望节省内存且不在意点之间的绝对距离，则可使用此语法。
- Vq = interp2(___,method) 指定备选插值方法："linear"、"nearest"、"cubic"、"makima" 或 "spline"。默认方法为 "linear"。
- Vq = interp2(___,method,extrapval) 还指定标量值或字符串 extrapval，如果该参数为标量值，则其会为处于样本点域范围外的所有查询点赋予该标量值，否则则采用该字符串指定的外插方法。



习题-2D 插值

- 对某NH_3气体，当其对比压力p_r=1.7476，T_r=1.106时，用普遍化压缩因子法求出其压缩因子Z
- 问题求解分析：普遍化压缩因子法的公式为$Z=Z_0+wZ_1$，查表可得NH_3的偏心因子w=0.25
- Z_0和Z_1需通过p_r和T_r的数据查表得到，如表1和2所示。

表1　Z_0值

T_r	P_r	
	1.5	2.0
1.1	0.458	0.3953
1.15	0.5798	0.476

表2　Z_1值

T_r	P_r	
	1.5	2.0
1.1	0.163	0.0698
1.15	0.1548	0.1667

- 从表1和表2可看出，在给定的临界温度（T_r=1.106）和临界压力（p_r=1.7476）下，Z_0和Z_1的数值在表中并不存在，需要用Matlab进行二维插值计算才能得到。

习题

- 解析计算：对行向量和列向量分别进行两次线性插值
- Mworks计算：
- w=0.25;
- x=[1.1 1.15]; y=[1.5 2.0]; % 输入表格中的Tr和pr
- zi0=[0.458 0.3953; 0.5798 0.476]; % 输入表格中对应的Tr和pr下的Z_0
- zi1=[0.1630 0.0698; 0.1548 0.1667]; % 输入表格中对应的Tr和pr下的Z_1
- z0=interp2(x,y,zi0,1.106,1.7476) % 用二维插值函数interp2计算Z_0
- z1=interp2(x,y,zi1,1.106,1.7476) % 用二维插值函数interp2计算Z_1
- Z=z0+w*z1
- --
- z0 =0.5083
- z1 =0.1540
- Z=0.5468
- 最终求得压缩因子Z=0.5468

曲线拟合

曲线拟合

- 将测量到的离散数据标出来，做成图。
- 对一个变量，可描成一条光滑的曲线，并将曲线用简单的数学表达式加以描述，这就是曲线拟合，又叫经验建模。
- 由已知数据确立经验或者半经验的数学模型，寻找相关变量之间的内在规律，又叫回归分析。

< 17 >

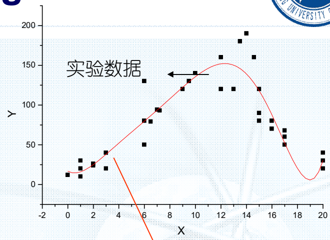
曲线拟合Curve Fitting

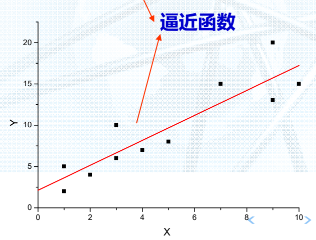
- 实验得到原始数据
- 研究数据的规律性
- 建立数学模型 y=f(x)
- 返回进行验证

- 最小二乘法（least square method）
- 最小二乘法（又称最小平方法）
- 数学优化技术，寻找数据的最佳拟合函数。
- 拟合标准：误差的平方和最小化

- 目标函数：

$$\min R = \min \sum_{i=1}^{n} [y_i - f(x_i)]^2$$





曲线拟合常用拟合方法

- 线性拟合:
  - 一元 $y = a + bx$
  - 多元 $y = a_0 + a_1x + a_2x_2 + \ldots a_nx_n$
- 非线性拟合方法
  - 二次多项式拟合：一元 $y = a_0 + a_1x + a_2x^2$
  - 高次多元拟合： $y = a_0 + a_1x_1 + a_2x_2^2 + \cdots + a_nx_n^m$
  - 非多项式形式 ⟶ 变量转换 ⟶ 多项式

$Nu = c_1 Re^{c_2} Pr^{c_3}$ $\ln(Nu) = \ln c_1 + c_2 \ln Re + c_3 \ln Pr$

$r = c/c_0 \exp(E/RT)$ $\ln r = \ln A + Bx$ $A = \frac{c}{c_0}$ $B = \frac{E}{R}$ $x = \frac{1}{T}$

< >

拟合解析求解

- 一元线性拟合：令 $f(x) = ax + b$ ，求拟合参数a, b
- 目标函数： $R = \sum_{i=1}^{n} [y_i - f(x_i)]^2 = \sum_{i=1}^{n} [y_i - ax_i - b]^2$

$$\left.\begin{aligned} \frac{dR}{da} &= -2\sum_{i=1}^{n}(y_i - ax_i - b)x_i = 0 \\ \frac{dR}{db} &= -2\sum_{i=1}^{n}(y_i - ax_i - b) = 0 \end{aligned}\right\} \text{线性方程组}$$

$$\left.\begin{aligned} \left(\sum_{i=1}^{n} x_i^2\right)a + \left(\sum_{i=1}^{n} x_i\right)b &= \sum_{i=1}^{n} x_i y_i \\ \left(\sum_{i=1}^{n} x_i\right)a + (n+1)b &= \sum_{i=1}^{n} y_i \end{aligned}\right\}$$

消元法
（参数或方程个数较少）
克莱姆(Gram) 法则
求参数*a, b*

< 20 >

用Excel 求解拟合问题

- 选择数据
- 插入散点图
- 选择趋势线
- 设置趋势线格式
- 得到拟合曲线

用Mwork求解：拟合函数fit

- 拟合函数：fit
- 函数库: TyCurveFitting
- fitobject = fit(fitType,x,y) 为 x,y 中的数据创建由 fitType 指定的模型的拟合。
- x：实验数据的自变量
- y: 实验数据的因变量
- fitType：拟合函数类型
  - 指定为字符串表示内置模型或函数表达式，或由 fittype 函数生成的 FitType。可以使用任何可以作为 fittype 第一个输入的变量作为 fit 的输入。

内置模型名称	描述
"poly1"	线性多项式曲线
"poly11"	线性多项式曲面
"poly2"	二次多项式曲线
"linearinterp"	分段线性插值
"cubicinterp"	分段三次插值
"smoothingspline"	平滑样条（曲线）
"lowess"	局部线性回归（曲面）

多项式模型名称与方程

曲线的多项式模型名称示例	方程
poly1	Y = p1*x+p2
直到 poly9	Y = p1*x^9+...+p10
["poly",10]	Y = p1*x^10+...+p11
...	...

用Mwork求解：plotfit函数

- 函数：plotfit #绘制 FitResult 对象
- 函数库: TyCurveFitting
- 语法
- H = plotfit(fun)
- H = plotfit(fun,x,y)
- 说明
- H = plotfit(fun) 在目前的坐标区绘制 FitResult 结构体且返回绘制对象的句柄 H。如果目前没有坐标区，fun 为来自 fit 函数的输出，则绘图将基于拟合点的坐标。
- H = plotfit(fun,x,y) 把 fun 和预测数据 x、响应数据 y 画在一起。

用Mwork求解：绘图函数figure/plot

- 函数：figure #创建图窗窗口
- 函数库: TyPlot
- figure()
- 使用默认属性值创建一个新的图窗窗口。
- 生成的图窗为当前图窗。

线型	说明	表示的线条
"-"	实线	———
"--"	虚线	- - - - -
":"	点线	········
"-."	点划线	-·-·-·-
"none"	无线条	无线条

- 函数：plot #绘制二维线图
- 函数库: TyPlot
- plot(X,Y)
- 创建 Y 中数据对 X 中对应值的二维线图
- plot(X,Y,fmt) #设置线型、标记符号和颜色
- plot(X1,Y1,fmt1,...,Xn,Yn,fmtn) #绘制多个X、Y对组的图，所有线条都使用相同的坐标区。设置每个线条的线型、标记符号和颜色
- plot(Y)
- 说明：如果 X 和 Y 都是向量，则它们的长度必须相同。plot 函数绘制 Y 对 X 的图。
- 如果 X 和 Y 均为矩阵，则它们的大小必须相同。plot 函数绘制 Y 的列对 X 的列的图。
- fmt表明线型、标记和颜色，指定为包含符号的字符串。符号可以按任意顺序显示。不需要同时指定所有三个特征（线型、标记和颜色）

用Mwork求解：绘图函数

- 标记和颜色

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
"red"	"r"	[1, 0, 0]	"#FF0000"	
"green"	不适用	[0, 0.5019607843137... 0]	"#008000"	
不适用	"g"	[0, 0.5, 0]	"#007F00"	
不适用	不适用	[0, 1, 0]	"#00FF00"	
"blue"	"b"	[0, 0, 1]	"#0000FF"	
"cyan"	不适用	[0, 1, 1]	"#00FFFF"	
不适用	"c"	[0, 0.75, 0.75]	"#00BFBF"	
"magenta"	不适用	[1, 0, 1]	"#FF00FF"	
不适用	"m"	[0.75, 0, 0.75]	"#BF00BF"	
"yellow"	不适用	[1, 1, 0]	"#FFFF00"	
不适用	"y"	[0.75, 0.75, 0]	"#BFBF00"	
"black"	"k"	[0, 0, 0]	"#000000"	
"white"	"w"	[1, 1, 1]	"#FFFFFF"	
"none"	不适用	不适用	不适用	无颜色

"d"		thin_diamond
"|"		vline
"_"		hline
"none"	无标记	none

标记	符号	说明
"."		point
","		pixel
"o"		circle
"v"		triangle_down
"^"		triangle_up
"<"		triangle_left
">"		triangle_right
"1"		tri_down
"2"		tri_up
"3"		tri_left
"4"		tri_right
"8"		octagon
"s"		square
"p"		pentagon
"P"		plus (filled)
"*"		star
"h"		hexagon1
"H"		hexagon2
"+"		plus
"x"		x
"X"		x (filled)
"D"		diamond

Mworks求解fit：多项式拟合程序版本1

- 如果没有预加载任何数据库，需要在程序前调用数据库：
- using TyCurveFitting
- using TyPlot
- using TyMath

- 如果预加载了数据库TyPlot、TyMath、只预加载了TyCurveFitting，没有预加载TyMachineLearning，则：
- t=[3:3:24;]
- y=[57.6,41.9,31,22.7,16.6,12.2,8.9,6.5];
- f1=fit("poly1",t,y)
- f2=fit("poly2",t,y)
- p1=f1.params #获取一阶多项式系数
- y1=polyval(p1,t) #求一阶多项式拟合结果
- p2=f2.params #获取二阶多项式系数
- y2=polyval(p2,t) #求二阶多项式拟合结果
- figure()
- plotfit(f1,t,y)
- hold("on") #保留当前坐标区中的绘图，从而使新添加到坐标区中的绘图不会删除现有绘图
- plotfit(f2,"r-.")
- hold("off")# 将保留状态设置为 "off"，从而使新添加到坐标区中的绘图清除现有绘图并重置所有的坐标区属性
- figure()
- plot(t,y,"r*",t,y1,"bo",t,y2,"k+")
- xlabel("t")
- ylabel("y")
- legend("data","1st-order","2nd-order")

Mworks求解fit：多项式拟合程序版本2

- 如果同时预加载了TyCurveFitting, TyMachineLearning这两个数据库
- t=[3:3:24;];
- y=[57.6,41.9,31,22.7,16.6,12.2,8.9,6.5];
- f1=TyCurveFitting.fit("poly1",t,y)
- f2=TyCurveFitting.fit("poly2",t,y)
- p1=f1.params #获取一阶多项式系数
- y1=polyval(p1,t) #求一阶多项式拟合结果
- p2=f2.params #获取二阶多项式系数
- y2=polyval(p2,t) #求二阶多项式拟合结果
- figure()
- TyCurveFitting.plotfit(f1,t,y)
- hold("on") #保留当前坐标区中的绘图，从而使新添加到坐标区中的绘图不会删除现有绘图
- TyCurveFitting.plotfit(f2,"r-.")
- hold("off")# 将保留状态设置为 "off"，从而使新添加到坐标区中的绘图清除现有绘图并重置所有的坐标区属性
- figure()
- plot(t,y,"r*",t,y1,"bo",t,y2,"k+")
- xlabel("t")
- ylabel("y")
- legend("data","1st-order","2nd-order")

注意数据库加载与否的不同

- 如果没有预加载函数库TyCurveFitting，需要添加语句：using TyCurveFitting #加载曲线拟合函数库
- 如果只预加载函数库TyCurveFitting，可不需要该语句。
- 因为fit函数在TyCurveFitting, TyMachineLearning这两个数据库里都有，如果这两个数据库同时预加载的话，调用该函数的时候就要变为：TyCurveFitting.fit
- 因为plotfit函数在TyCurveFitting, TyMachineLearning这两个数据库里都有，如果这两个数据库同时预加载的话，调用该函数的时候就要变为：TyCurveFitting.plotfit

Mworks求解fit：多项式拟合结果

y1=
49.0583
42.0917
35.1250
28.1583
21.1917
14.2250
7.2583
0.2917

y2=
56.2833
43.1238
32.0286
22.9976
16.0310
11.1286
8.29048
7.51667

p1=
-2.322
56.025

p2=
0.1147
-5.419
71.507

注意：Vector与Matrix的不同

- Vector与Matrix的不同

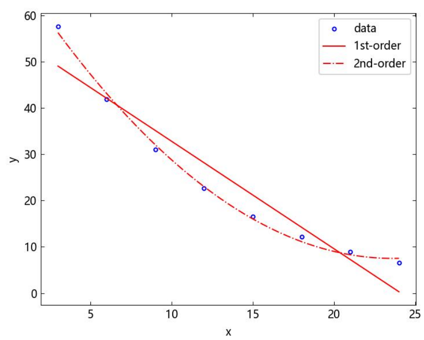
x=[1:1:10;]	x=[1,2,3,4,5,6,7,8,9,10]	x=[1:1:10...]
10-element Vector{Int64}:	10-element Vector{Int64}:	10-element Vector{Int64}:
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

x=[1 2 3 4 5 6 7 8 9 10]
1×10 Matrix{Int64}:
1 2 3 4 5 6 7 8 9 10

x=[1 2; 3 4]
2×2 Matrix{Int64}:
1 2
3 4

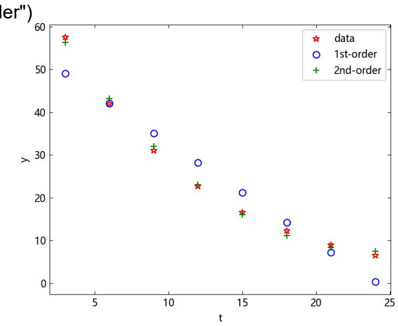
Mworks求解fit：多项式拟合结果 画图1

- figure()
- TyCurveFitting.plotfit(f1,t,y)
- hold("on") TyCurveFitting.plotfit(f2,"r-.")
- hold("off")



Mworks求解fit：多项式拟合结果 画图2

```
figure()
plot(t,y,"r*",t,y1,"bo",t,y2,"k+")
xlabel("t")
ylabel("y")
legend("data","1st-order","2nd-order")
```



用Mwork求解：多项式拟合函数polyfit

$$p(x) = p_1x^n + p_2x^{n-1} + \ldots + p_nx + p_{n+1}$$

- 多项式曲线拟合：polyfit
- 函数库: TyMath
- p,S=polyfit(x,y,n)#返回次数为n的多项式p(x)的系数，该阶数是y中数据的最佳拟合（在最小二乘方式中）以及返回一个结构体S，后者可用作polyval的输入来获取误差估计值。p的系数按降幂排列， p 的长度为 n+1 。
- t=[3:3:24];
- y=[57.6,41.9,31,22.7,16.6,12.2,8.9,6.5];
- p1,S1=polyfit(t,y,1)
- p2,S2=polyfit(t,y,2)
- y1=polyval(p1,t)
- y2=polyval(p2,t)
- 结果:
- p1=[-2.322, 56.025]
- p2=[0.1147, -5.4187, 71.5071]

例. 在研究某单分子化学反应速度时, 得到下列数据:

i	1	2	3	4	5	6	7	8
t_i	3	6	9	12	15	18	21	24
y_i	57.6	41.9	31.0	22.7	16.6	12.2	8.9	6.5

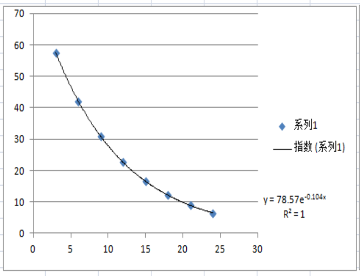
- 其中t表示从实验开始算起的时间, y表示时刻 t反应物的量。
- 试根据上述数据求出化学反应速率的经验公式
- 由化学反应速度的理论知, 经验公式 $y = f(t) = ke^{mt}$
- 图解法: $\ln y = mt + \ln k \longrightarrow Y = aX + b$

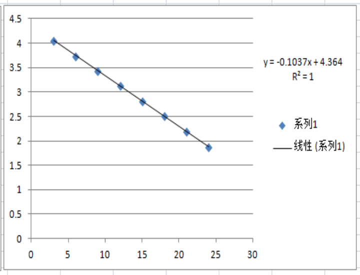
$$令\ Y = \ln y,\ X = \tau,\ a = m,\ b = \ln k$$

- 从而求得k，m

Excel 求解

	A	B	C	D	E	F	G	H	I
1	次数	1	2	3	4	5	6	7	8
2	time	3	6	9	12	15	18	21	24
3	y	57.6	41.9	31	22.7	16.6	12.2	8.9	6.5
4	Y	4.053523	3.735286	3.433987	3.122365	2.809403	2.501436	2.186051	1.871802





	A	B
24	a=m	-0.1037
25	b=lnk	4.363
26	k	78.49226

Mworks求解fit：指数函数拟合

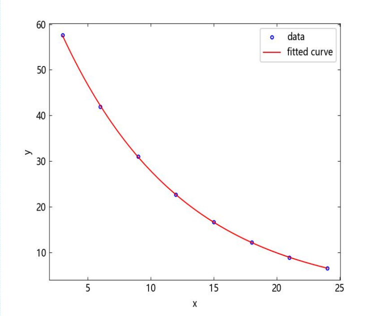
$$y = ae^{bx}$$

- 如果拟合函数为指数，使用 fit 函数对数据进行指数模型拟合。
- 指数库模型是 fit 和 fittype 函数的输入参量。指定模型类型 "exp1"
- f = fit("exp1",x,y)
- plotfit(f,x,y)
- 或者以交互方式拟合：
  - 通过在 Syslab 命令行中输入 curveFitter() 打开曲线拟合器。或者，在 APP 选项卡上单击曲线拟合器。
  - 在曲线拟合器中，选择曲线数据。在曲线拟合器选项卡的数据部分中，单击选择数据。在选择拟合数据对话框中，选择 X 数据和 Y 数据，或仅选择 Y 数据（相对于索引绘图）。
  - 单击拟合类型部分中的箭头以打开库，然后单击回归模型组中的指数。

Mworks求解fit：指数函数拟合

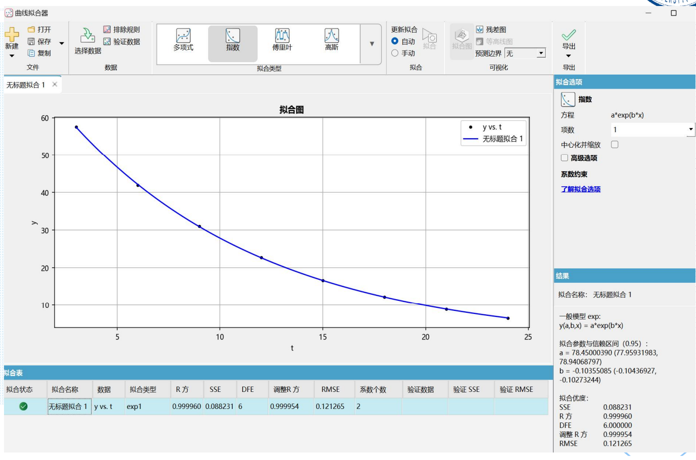
- t=[3:3:24;];
- y=[57.6,41.9,31,22.7,16.6,12.2,8.9,6.5];
- f = fit("exp1",t,y)
- p=f.params #获取拟合参数
- plotfit(f,t,y)

- 结果：
- p=
- 78.4500
- -0.1036



$$y = 78.45e^{-0.1036t}$$

交互界面：曲线拟合器



用Mworks求解：最小二乘非线性拟合函数

- lsqcurvefit：用最小二乘求解非线性曲线拟合（数据拟合）问题
- 函数库: TyOptimization
- x,resnorm,residual,exitflag,output,lambda,jacobian= lsqcurvefit(fun,x0,xdata,ydata,lb,ub,A,b,Aeq,beq)
- 从 x0 开始，求取合适的系数 x，使非线性函数 fun(x,xdata) 对数据 ydata 的拟合最佳（基于最小二乘指标）。
- ydata 必须与 fun 返回的向量（或矩阵）F 大小相同

找到求解以下问题的系数 x

$$\min_x ||F(x, xdata) - ydata||_2^2 = \min_x \sum_i (F(x, xdata_i) - ydata_i)^2$$

给定输入数据 xdata，观察到的输出 ydata，其中 xdata 和 ydata 是矩阵或向量，F (x, xdata) 是与 ydata 大小相同的矩阵值或向量值函数。

（可选）x 的分量需满足以下约束

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

参数 x、lb 和 ub 为向量。

- x - 最优系数
- resnorm - 在 x 处的残差的 2-范数平方值：sum(abs2,(fun(x,xdata)-ydata))
- residual - 在解 x 处的残差 fun(x,xdata)-ydata 的值
- exitflag - 描述退出条件的值
- 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
- jacobian - fun 在解 x 处的雅可比矩阵

用Mworks求解：最小二乘非线性拟合函数

- lsqlin: 求解约束的线性最小二乘问题
- Solve constrained linear least-squares problems
- x= lsqlin(C,d,A,b)
- x= lsqlin (C,d,A,b) attempts to solve the least-squares problem
- min 0.5*(NORM(C*x-d)).^2 subject to A*x <= b

- lsqnonlin: 求解非线性最小二乘曲线拟合问题
- Solves nonlinear least-squares curve fitting problems

- lsqnonneg: 求解非负的最小二乘约束问题
- Solve nonnegative least-squares constraints problem

Mworks 求解：lsqcurvefit

- 新建Julia脚本文件，定义函数文件'cherate.jl'，要运行一下。
- function cherate(x,xdata)
-     F = x[1].*exp.(x[2].*xdata);
-     return F
- end
- --
- 定义主文件（Julia脚本）：ratefit.jl
- using TyOptimization
- t=[3:3:24;];
- y=[57.6,41.9,31,22.7,16.6,12.2,8.9,6.5];
- x0=[10,-1];
- x,resnorm,residual,exitflag,output,lambda,jacobian= lsqcurvefit(cherate,x0,t,y);
- x
- --
- 运行主文件ratefit.jl，得到计算结果
- x=[78.4500, -0.1036]

$$y = 78.45e^{-0.1036t}$$

Mworks 求解：lsqcurvefit

- 另一种建立函数的方法：
- using TyOptimization
- t=[3:3:24;];
- y=[57.6,41.9,31,22.7,16.6,12.2,8.9,6.5];
- x0=[10,-1];
- fun = (x,xdata)->x[1].*exp.(x[2].*xdata)
- x,resnorm,residual,exitflag,output,lambda,jacobian= lsqcurvefit(fun,x0,t,y);
- x或print(x)
- 结果：x=[78.4500, -0.1036]

$$y = 78.45e^{-0.1036t}$$

结论

- 用Mworks对数据进行插值和拟合
- 用Excel对数据进行拟合