数值计算与数据处理

第9讲

数值计算概述

数值计算的必要性：

- 实际工程问题的微分方程往往很复杂，不能用符号运算获得解析解。
- 对于实际工程问题，满足工程精度要求的若干个点的近似解已经够用，没有必要计算精确解。

数值计算概述

- 1. 数值计算方法
- 2. 微分方程的数值计算
- 3. 数值积分
- 4.数据拟合与插值

9.1 微分方程的数值计算方法

设 $x_{i+1}-x_i=h, i=0,1,2,\cdots,n-1,\cdots$ ，可以用以下离散化方法

求解微分方程:
$$\begin{cases} y'=f(x,y) \\ y(x_0)=y_0 \end{cases}$$

1、用差商代替导数(欧拉法)

若步长h较小，则有

$$y'(x)\approx\frac{y(x+h)-y(x)}{h}$$

故有公式:

$$\begin{cases} y_{i+1}=y_i+hf(x_i,y_i) \\ y_0=y(x_0) \end{cases} \quad \mathrm{i}=0,1,2,\cdots,\mathrm{n}-1$$

9.1 微分方程的数值计算方法

2、使用数值积分(改进欧拉法)

对方程y'=f(x,y), 从x_i到x_{i+1}积分，利用梯形公式，有：

$$y(x_{i+1})-y(x_i)=\int_{x_i}^{x_{i+1}} f(t,y(t))dt \approx \frac{x_{i+1}-x_i}{2}[f(x_i,y(x_i))+f(x_{i+1},y(x_{i+1}))]$$

故有公式：

$$\begin{cases} y_{i+1}=y_i+\dfrac{h}{2}[f(x_i,y_i)+f(x_{i+1},y_{i+1})] \\ y_0=y(x_0) \end{cases}$$

在实际应用时，结合欧拉公式使用：

$$\begin{cases} y_{i+1}^{(0)}=y_i+hf(x_i,y_i) \\ y_{i+1}^{(k+1)}=y_i+\dfrac{h}{2}[f(x_i,y_i)+f(x_{i+1},y_{i+1}^{(k)})] \quad k=0,1,2,\cdots \end{cases}$$

对于给定的精度 ε，当满足$\left|y_{i+1}^{(k+1)}-y_{i+1}^{(k)}\right|<\varepsilon$时，取 $y_{i+1}=y_{i+1}^{(k+1)}$

然后继续下一步 y_{i+2} 的计算

9.1 微分方程的数值计算方法

3、使用泰勒公式

包括龙格-库塔法(Runge-Kutta)、线性多步法等方法。

根据一阶精度的拉格朗日中值定理，对于微分方程有：

$$y'=f(x,y),y(i+1)=y(i)+h*K1,K1=f(x_i,y_i)$$

用点x_i处的斜率K1与右端点x_{i+1}处的斜率K2的算术平均值作为平均斜率K^*的近似值，得到二阶精度的改进拉格朗日中值定理：

y(i+1)=y(i)+[h*(K1+ K2)/2]

$K1=f(x_i,y_i)$

K2=f(x(i)+h,y(i)+h*K1)

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}K_1) \\ K_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}K_2) \\ K_4 = f(x_i + h, y_i + hK_3) \end{cases}$$

9.1 微分方程的数值计算方法

数值公式的精度

- 当一个数值公式的截断误差可表示为$O(h^{k+1})$时(k为正整数，h为步长)，称它是一个k阶公式。
- k越大，则数值公式的精度越高。
- 欧拉法是一阶公式，改进的欧拉法是二阶公式。
- 龙格-库塔法有二阶公式和四阶公式。
- 线性多步法有四阶阿达姆斯(Adams)外插公式和内插公式。

9.1 微分方程的数值计算

[t, x]=solver(odefun, ts, x_0, options）

- t：自变量值
- x：函数值
- solver：ode45、ode23、ode113、ode15s、ode23s
- odefun：由待解方程写成的m-文件名
- ts：ts=[t_0，t_f]，t_0、t_f为自变量的初值和终值
- x_0：函数的初值

ode23：组合的2/3阶龙格-库塔-芬尔格算法

ode45：运用组合的4/5阶龙格-库塔-芬尔格算法

options：用于设定误差限(缺省时设定相对误差10^{-3}, 绝对误差10^{-6}),
命令为：options=odeset('reltol',rt,'abstol',at),
rt，at：分别为设定的相对误差和绝对误差.

9.1 微分方程的数值计算

微分方程(组)的几种类型：

- **显式微分方程**

$$y' = f(t, y) \quad y(t_0) = y_0$$

刚性方程：ode15s、ode23s、ode23b、ode23t

非刚性方程： ode45、ode23、ode113

- **微分代数方程**

$$M(t, y)y' = f(t, y) \quad y(t_0) = y_0$$

- **延时微分方程(dde23)**

$$y = f(t, y(t), y(t-\tau_1), \cdots, y(t-\tau_k))$$

9.1 微分方程的数值计算

注意：

使用Matlab软件求数值解时，高阶微分方程必须等价地变换成一阶微分方程组。

$$y^{(n)} = f\left(t, y, y' \cdots y^{(n-1)}\right)$$

$$y'' = \frac{1}{5}\sqrt{1 + y'^2}$$

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ &\vdots \\ y_n' &= f\left(t, y_1, y_2 \cdots y_{n-1}\right) \end{aligned}$$

$$\begin{cases} y'_1 = y_2 \\ y_2' = \dfrac{1}{5}\sqrt{1 + y_1^2} \Big/ (1 - t) \end{cases}$$

9.1 微分方程的数值计算

注意：

➢在解n个未知函数的方程组时，x0和x均为n维向量，m文件中的待解方程组应以x的分量形式处理。

1.建立m-文件myfun.m

```
function dy=myfun(t,y)
dy=zeros(2,1);
dy(1)=y(2);
dy(2)=1/5*sqrt(1+y(1)^2)/(1-t);
```

$$\begin{cases} y'_1 = y_2 \\ y_2' = \dfrac{1}{5}\sqrt{1+y_1^2} \Big/ (1-t) \end{cases}$$

2. 初始条件：x0=0,xf=1

主程序

```
[x,y]=ode15s('myfun',[x0 xf],[0 0]);
plot(x,y(:,1))
```

例：二阶微分方程的降阶

$$\begin{cases} \dfrac{d^2x}{dt^2} - 1000\left(1 - x^2\right)\dfrac{dx}{dt} - x = 0 \\ x(0) = 2, x'(0) = 0 \end{cases}$$

降阶处理：

令 $y_1=x$，$y_2=y_1'$

则微分方程变为一阶微分方程组

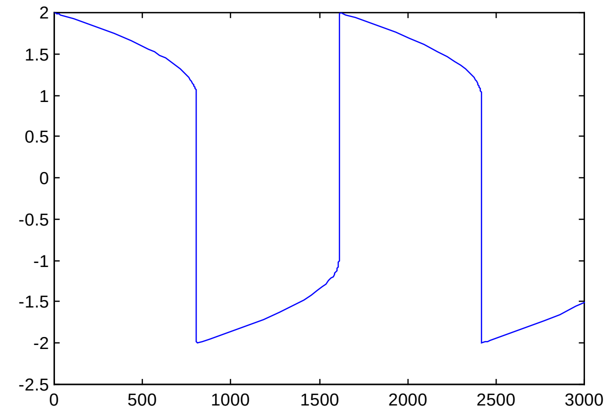
$$\begin{cases} y_1' = y_2 \\ y_2' = 1000(1 - y_1^2)y_2 - y_1 \\ y_1(0) = 2, y_2(0) = 0 \end{cases}$$

例：二阶微分方程

$$\begin{cases} y_1' = y_2 \\ y_2' = 1000(1-y_1^2)y_2 - y_1 \\ y_1(0) = 2,\ y_2(0) = 0 \end{cases}$$

1、建立m-文件vdp1000.m如下：

```
function dy=vdp1000(t,y)
dy=zeros(2,1);
dy(1)=y(2);
dy(2)=1000*(1-y(1)^2)*y(2)-y(1);
```



2、取t0=0，tf=3000，输入命令：

```
[T,Y]=ode15s('vdp1000',[0 3000],[2 0]);
plot(T,Y(:,1),'-')
```

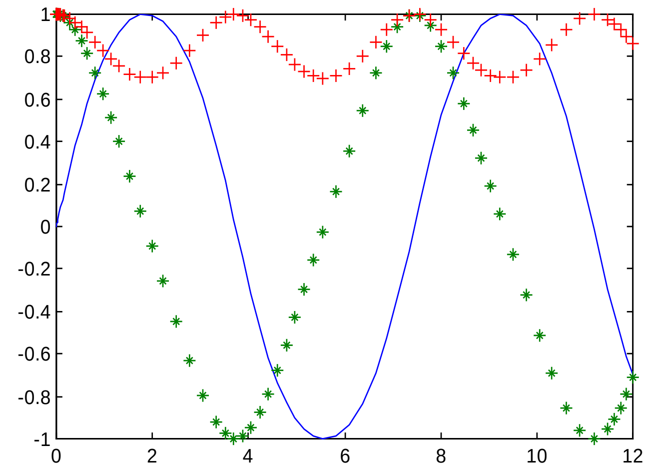
例：微分方程组

解微分方程组

$$\begin{cases} y_1' = y_2 y_3 \\ y_2' = -y_1 y_3 \\ y_3' = -0.51 y_1 y_2 \\ y_1(0)=0,\ y_2(0)=1,\ y_3(0)=1 \end{cases}$$

1、建立m-文件rigid.m如下：

```
function dy=rigid(t,y)
dy=zeros(3,1);
dy(1)=y(2)*y(3);
dy(2)=-y(1)*y(3);
dy(3)=-0.51*y(1)*y(2);
```



2、取t0=0，tf=12，输入命令：

```
[T,Y]=ode45('rigid',[0 12],[0 1 1]);
plot(T,Y(:,1),'-',T,Y(:,2),'*',T,Y(:,3),'+')
```

例：微分代数方程组

$$\begin{cases} (m_1+m_2)x''-(m_2L\sin\theta)\theta''=m_2L(\theta')^2\cos\theta \\ (m_1+m_2)y''-(m_2L\cos\theta)\theta''=m_2L(\theta')^2\sin\theta-(m_1+m_2)g \\ (-L\sin\theta)x''+(L\cos\theta)y''+(L^2\theta')\theta''=-gL\cos\theta \end{cases}$$

$$M(t,q)q'=f(t,q) \quad q(t_0)=q_0 \qquad q=\begin{bmatrix}x & x' & y & y' & \theta & \theta'\end{bmatrix}^{\mathrm{T}}$$

$$\begin{cases} x'=x' \\ (m_1+m_2)x''-(m_2L\sin\theta)\theta''=m_2L(\theta')^2\cos\theta \\ y'=y' \\ (m_1+m_2)y''-(m_2L\cos\theta)\theta''=m_2L(\theta')^2\sin\theta-(m_1+m_2)g \\ \theta'=\theta' \\ (-L\sin\theta)x''+(L\cos\theta)y''+(L^2\theta')\theta''=-gL\cos\theta \end{cases}$$

例：微分代数方程组

```
function M=mass(t,y,m1,m2,L,g)
M=zeros(6,6);
M(1,1)=1;
M(2,2)=m1+m2;
M(2,6)=-m2*L*sin(y(5));
M(3,3)=1;
M(4,4)=m1+m2;
M(4,6)=m2*L*cos(y(5));
M(5,5)=1;
M(6,2)=-L*sin(y(5));
```

$$M=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & m_2 L\sin\theta & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & m_1+m_2 & 0 & m_2 L\cos\theta\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & -L\sin\theta & 0 & L\cos\theta & 0 & L^2\theta'\end{bmatrix}$$

$$\begin{cases} y(1)=x\\ y(2)=x'\\ y(3)=y\\ y(4)=y'\\ y(5)=\theta\\ y(6)=\theta'\end{cases}$$

例：微分代数方程组

```
function dydt=massode(t,y,m1,m2,L,g)
dydt=[
y(2);
m2*L*y(6)^2*cos(y(5));
y(4);
m2*L*y(6)^2*sin(y(5)-(m1+m2))*g;
y(6);
-g*L*cos(y(5));
];
```

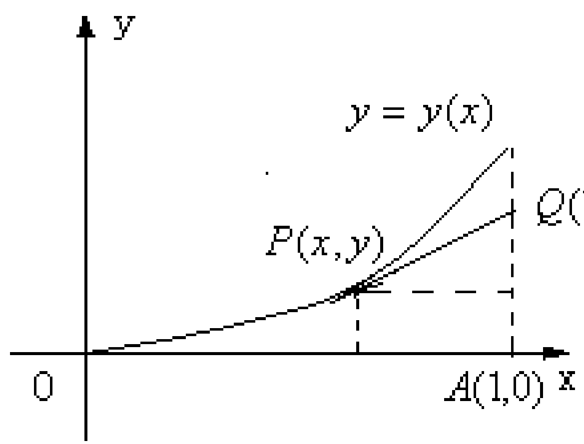
$$f(t,y)=\begin{bmatrix} x' \\ m_2L(\theta')^2\cos\theta \\ y' \\ m_2L(\theta')^2\sin\theta-(m_1+m_2)g \\ \theta' \\ -gL\cos\theta \end{bmatrix}$$

```
m=0.1;m2=0.1;L=1;g=9.81;
tspan=linspace(0,4,25);
y0=[0 4 2 20 –pi/2 2];
options=odeset('Mass',@mass);
options=odeset('RelTol',1e-4);
[t y]=ode45(@massode,tspan,y0,options,m1,m2,L,g);
```

$$\begin{cases} y(1)=x \\ y(2)=x' \\ y(3)=y \\ y(4)=y' \\ y(5)=\theta \\ y(6)=\theta' \end{cases}$$

例：导弹跟踪问题

设位于坐标原点的甲舰向位于x轴上点A(1, 0)处的乙舰发射导弹，导弹头始终对准乙舰.如果乙舰以最大的速度v_0(是常数)沿平行于y轴的直线行驶，导弹的速度是$5v_0$，求导弹运行的曲线方程。又乙舰行驶多远时，导弹将它击中?



例：解析法求导弹跟踪问题

假设导弹在t时刻的位置为P(x(t),y(t)),乙舰位于Q(1,v_0t)

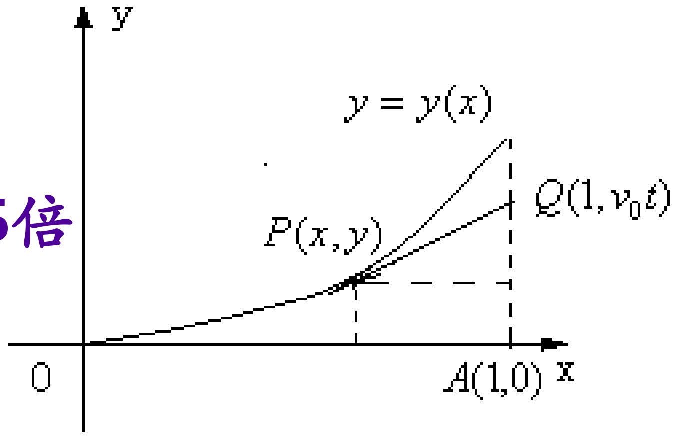
由于导弹头始终对准乙舰，故此时直线PQ就是导弹的轨迹曲线弧OP在点P处的切线

即有 $$y' = \frac{v_0 t - y}{1-x}$$

即 $$v_0 t = (1-x)y' + y \qquad (1)$$

又根据题意，弧OP的长度为$|AQ|$ 的5倍

即 $$\int_0^x \sqrt{1+y'^2}\,dx = 5v_0 t \qquad (2)$$



例：解析法求导弹跟踪问题

由(1),(2)消去t整理得方程:

$$(1-x)y''=\frac{1}{5}\sqrt{1+y'^2} \qquad (3)$$

初值条件为： $y(0)=0,\ y'(0)=0$

方程的解即为导弹的运行轨迹

$$y=-\frac{5}{8}(1-x)^{\frac{4}{5}}+\frac{5}{12}(1-x)^{\frac{6}{5}}+\frac{5}{24}$$

当 $x=1$ 时，即 $y=\frac{5}{24}$ 当乙舰航行到点 $(1,\frac{5}{24})$ 处时被导弹击中，被击中时间为 $t=\frac{y}{v_0}=\frac{5}{24v_0}$ 。若 $v_0=1$ 则在 $t=0.21$ 处被击中。

例：数值法求导弹跟踪问题

令y1=y,y2=y1'，将方程（3）化为一阶微分方程组

$$(1-x)y''=\frac{1}{5}\sqrt{1+y^2}$$

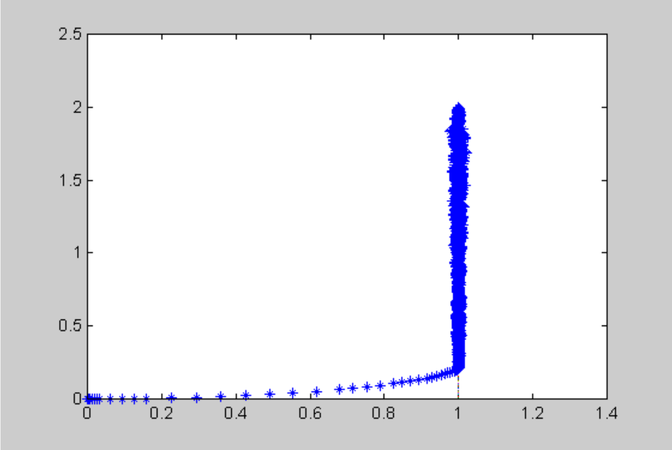
$$\begin{cases} y'_1 = y_2 \\ y_2' = \frac{1}{5}\sqrt{1+y_1^2}\,/(1-x) \end{cases}$$

1.建立m-文件eq1.m

```
function dy=eq1(x,y)
dy=zeros(2,1);
dy(1)=y(2);
dy(2)=1/5*sqrt(1+y(1)^2)/(1-x);
```

2. 取x0=0，xf=0.9999，建立主程序ff6.m如下：

```
x0=0，xf=0.9999
[x,y]=ode15s('eq1',[x0 xf],[0 0]);
plot(x,y(:,1), 'b.')
hold on
y=0:0.01:2;
plot(1,y,'b*')
```



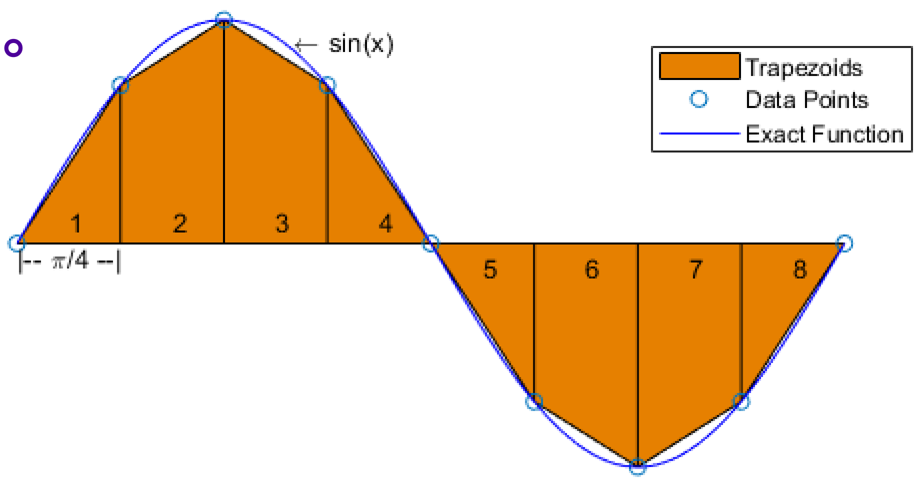
9.3 数值积分

trapz函数通过梯形法计算近似积分，调用格式为：

- Q = trapz(Y)

■ 如果Y为向量，计算由坐标(i,Y(i))确定一组数据点构成的曲线积分。

■ 如果Y为矩阵，计算由坐标(i,Y(i,1:end))确定一组数据点构成的曲线积分。即对每列求积分并返回积分值的行向量。



9.3 数值积分

- Q = trapz(X,Y)

- 计算由坐标(X,Y)确定一组数据点构成的曲线积分。
- X 指定数据点的 x 坐标，X必须与 Y 的积分维度大小相同。

例如：

```
X = [1 2.5 7 10];
Y = [5.2 7.7 9.6 13.2];
trapz(X,Y)
ans=
82.8
```

9.3 数值积分

- Q = trapz(Y,dim)

  其中Y为矩阵。

- Q = trapz(X,Y,dim)

  其中X为向量，Y为矩阵。

沿维度 dim 求积分。dim为1表示按列积分，返回行向量。dim为2表示按行积分，返回列向量。

例如：

```
X = [1 2.5 7 10];
Y = [5.2 7.7 9.6 13.2;
     4.8 7.0 10.5 14.5;
     4.9 6.5 10.2 13.8];
Q1 = trapz(X,Y,2)
Q2 = trapz(X,Y',1)
```

9.4 数据插值与拟合

- 数据插值

通过原始数据计算一些新的离散数据点。

- 曲线拟合

从一些离散的数据中推导出两者之间的数学解析关系。

函数	说明
interp1	一维插值
polyfit	最小二乘法曲线拟合
polyval	多项式函数值计算

9.4.1 数据插值

1、一维插值

interp1(x,y,xi,插值算法)

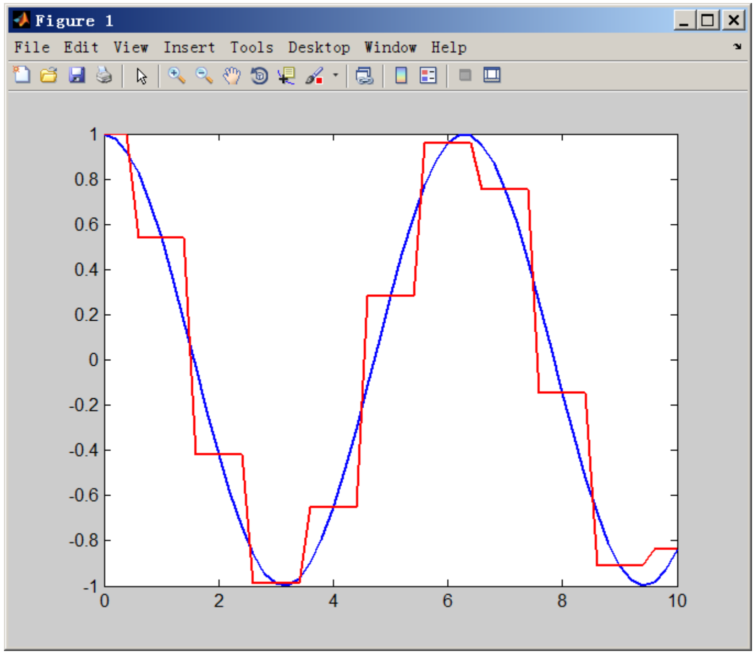
yi = interp1(x, y, xi, method)

其中：

* x和y为向量，为原始数据(自变量和对应的函数值)。
* xi为需要计算的插值点
* method指定插值计算使用的算法，可以为 nearest (最近点等值) 、linear(线性插值，即直线)、spline(三次分段样条插值)。

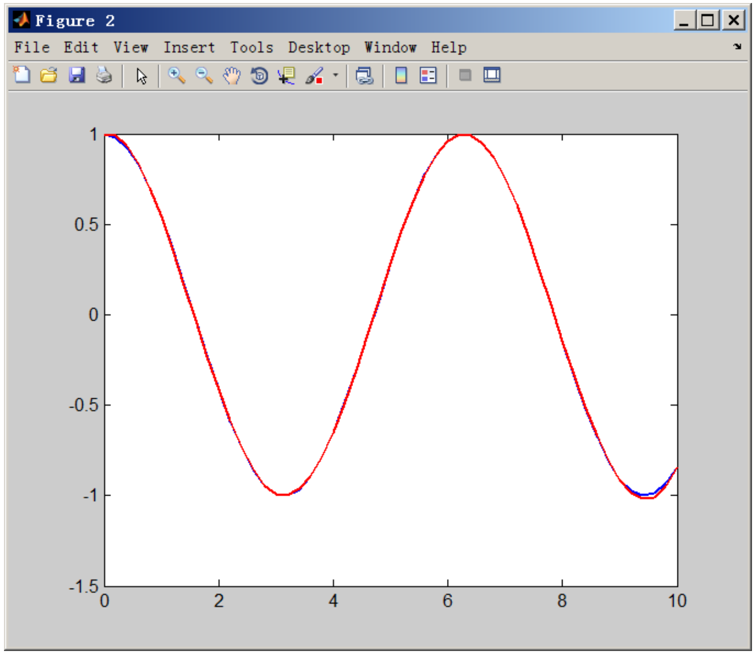
9.4.1 数据插值

```
x = 0:10;
y = cos(x);
xi = 0:0.2:10;
yi=cos(xi);
yin = interp1(x,y,xi,'nearest');
plot(xi,yi,'LineWidth',1.5)
hold on
plot(xi,yin,'-r','LineWidth',1.5)
```



9.4.1 数据插值

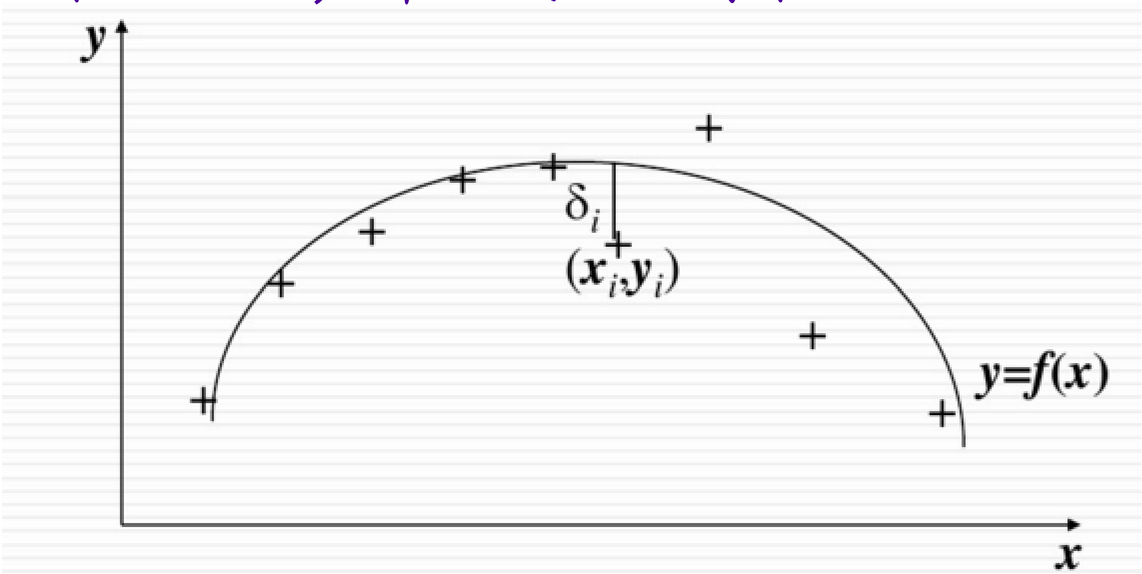
```
x = 0:10;
y = cos(x);
xi = 0:0.2:10;
yi=cos(xi);
yis = interp1(x,y,xi, 'spline');
plot(xi,yi,'LineWidth',1.5)
hold on
plot(xi,yis,'-r','LineWidth',1.5)
```



9.4.2 数据拟合

➢ 曲线拟合的目的

已知一组数据(二维)，即平面上n个点，$(x_i, y_i)\ i=1,2,\cdots,n$ 寻求一个函数(曲线)y=f(x)，使得f(x)在某种准则下雨所有数据点最为接近，即曲线拟合得最好。



δ_i 为点 (x_i, y_i) 与曲线 $y=f(x)$ 的距离

9.4.2 数据拟合

➢ **曲线拟合的常用解法——线性最小二乘法**

第一步：先选定一组函数，令 $r_1(x), r_2(x), \cdots, r_m(x), m < n$

$$f(x) = a_1 r_1(x) + a_2 r_2(x) + \cdots + a_m r_m(x)$$

其中 $a_1, a_2, \cdots, a_m$ 为待定系数

第二步：确定的准则(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

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

$$= \sum_{i=1}^{n} \left[\sum_{k=1}^{m} a_k r_k(x_i) - y_i \right]^2$$

问题归结为，求 $a_1, a_2, \cdots, a_m$ 使 $J(a_1, a_2, \cdots, a_m)$ 最小。

9.4.2 数据拟合

多项式拟合

$$y = a_0 + a_1 t + a_2 t^2$$

$$y = a_0 + a_1 e^{-t} + a_2 t e^{-t}$$

曲线拟合就是求得多项式的系数。

三组数据：(0.3,0.82)、(0.8,1.14)、(1.1,1.25)

```
t = [0.3 0.8 1.1]';
Y = [0.82 1.14 1.25]';
T=[ones(3,1);t;t.^2];
A=inv(T)*Y
```

$$\begin{bmatrix} 1 & 0.3 & 0.3^2 \\ 1 & 0.8 & 0.8^2 \\ 1 & 1.1 & 1.1^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0.82 \\ 1.14 \\ 1.25 \end{bmatrix}$$

9.4.2 数据拟合(确定多项式系数)

$$y = p_n x^n + p_{n-1} x^{n-1} + \ldots + p_1 x + p_0$$

p = polyfit(x,y,n)

其中，x和y为参与曲线拟合计算的原始数据

n为进行拟合计算的多项式次数

函数的返回值是向量形式的多项式的系数。

```
x=0:0.2:10;
y=0.25*x+20*sin(x);
p5=polyfit(x,y,5)
p5 =
    0.0161   -0.6181    7.6911  -37.4106   60.9743   -8.4150
```

$$y = 0.0161x^5 - 0.6181x^4 + 7.6911x^3 - 37.4106x^2 + 60.9743x - 8.415$$

9.4.2 数据拟合(确定拟合点的函数值)

如何计算拟合后各点的函数值?

y = polyval(p,x)

其中，p为多项式的系数，x为变量的数值

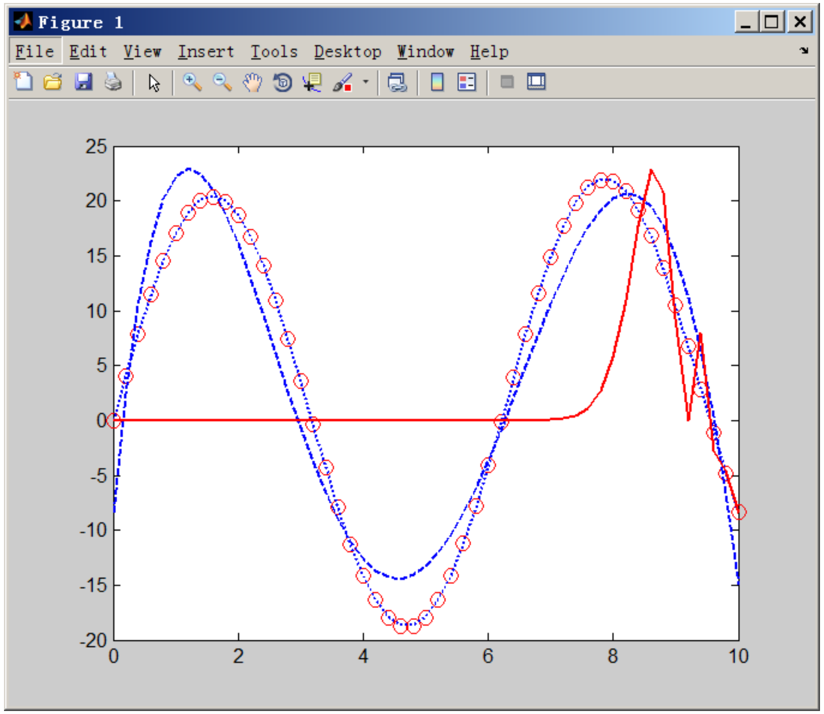
y为变量x对应的函数值。

```
x=0:0.2:10;
y=0.25*x+20*sin(x);
%5阶多项式拟合
p5=polyfit(x,y,5);
y5=polyval(p5,x);
```

例：数据拟合的阶数选择

```
x=0:0.2:10;
y=0.25*x+20*sin(x);
%5阶多项式拟合
p5=polyfit(x,y,5);
y5=polyval(p5,x);
%8阶多项式拟合
p8=polyfit(x,y,8);
y8=polyval(p8,x);
%60阶多项式拟合
p60=polyfit(x,y,60);
y60=polyval(p60,x);
plot(x,y,'ro',x,y5,'b--',x,y8,'m:',x,y60,'r-')
```

多项式拟合的阶数必须适中，不是阶数越高精度越高。



例：数据拟合精度分析

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

```
n=length(y);
mse5=sum((y-y5).^2)/n
mse5 =
  13.1029
mse8=sum((y-y8).^2)/n
mse8 =
   0.0036
```

```
x0=[0,3,5,7,9,11,12,13,14,15];
y0=[0,1.2,1.7,2.0,2.1,2.0,1.8,1.2,1.0,1.6];
```