

第六章 数值微分

6.1 引言

6.2 导数的近似值

6.3 插值型求导公式

6.4 数值差分公式

6.1 引言

回顾

数值微分是用离散方法近似地计算函数在某点的导数值。

在微分学中，求函数 $f(x)$ 的导数 $f'(x)$ 通常比 $f(x)$ 复杂的多。另外，求数值导数也是实际问题经常遇到的，特别当该函数本身未知，但又需要对其求导数时，数值微分方法显得更为重要。

此外，在实际问题中，往往会遇到某函数 $f(x)$ 是用表格表示的，用通常的导数定义无法求导，因此要寻求其他方法近似求导。

求解数值导数的公式对开发求解常微分方程和偏微分方程边值问题的算法非常重要。

6.3 数值差分公式

最简单的数值微分是用向前差分近似代替导数（如果 $f(x)$ 在点 x 的右边的值可计算），即

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

同样，也可用向后差分近似代替导数（如果 $f(x)$ 在点 x 的左边的值可计算），即

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h}$$

或中心差分的方法（如果 $f(x)$ 在点 x 的左右两边的值都可计算），即

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

可以看出中心差分是向前差分和向后差分的算术平均值。上述三种方法的截断误差分别为 $O(h)$ 、 $O(h)$ 和 $O(h^2)$ 。

6.3 插值型求导公式

回顾

函数 $f(x)$ 的定积分可以用插值多项式 $P(x)$ 的定积分来近似计算, 同样, 函数 $f(x)$ 的导数也可以用插值多项式 $P(x)$ 的导数来近似代替, 即

$$f'(x) \approx P'(x) \quad (11)$$

这样建立的数值微分公式, 统称为插值型求导公式。

应当指出的是即使 $P(x)$ 与 $f(x)$ 处处相差不多, 但 $P'(x)$ 与 $f'(x)$ 在某些点仍然可能出入很大, 因而在使用插值求导公式时, 要注意误差的分析。

第六章 微分问题的计算机求解

- 微分问题的解析解
- 数值微分
- 函数的级数展开与级数求和问题求解

6.1 函数导数的解析解

- 函数的导数和高阶导数

$$\frac{df(x)}{dx}$$

$$\frac{d^n f(x)}{dx^n}$$

– 格式: `y=diff(fun,x)` %求导数

`y= diff(fun,x,n)` %求n阶导数

- 例: 函数 $f(x) = \sin x / (x^2 + 4x + 3)$ 求出 $\frac{d^4 f(x)}{dx^4}$

一阶导数:

```
>> syms x; f=sin(x)/(x^2+4*x+3);
```

```
>> f1=diff(f); pretty(f1)
```

$$\frac{\cos(x)}{2x^2 + 4x + 3} - \frac{\sin(x)(2x + 4)}{(2x^2 + 4x + 3)^2}$$

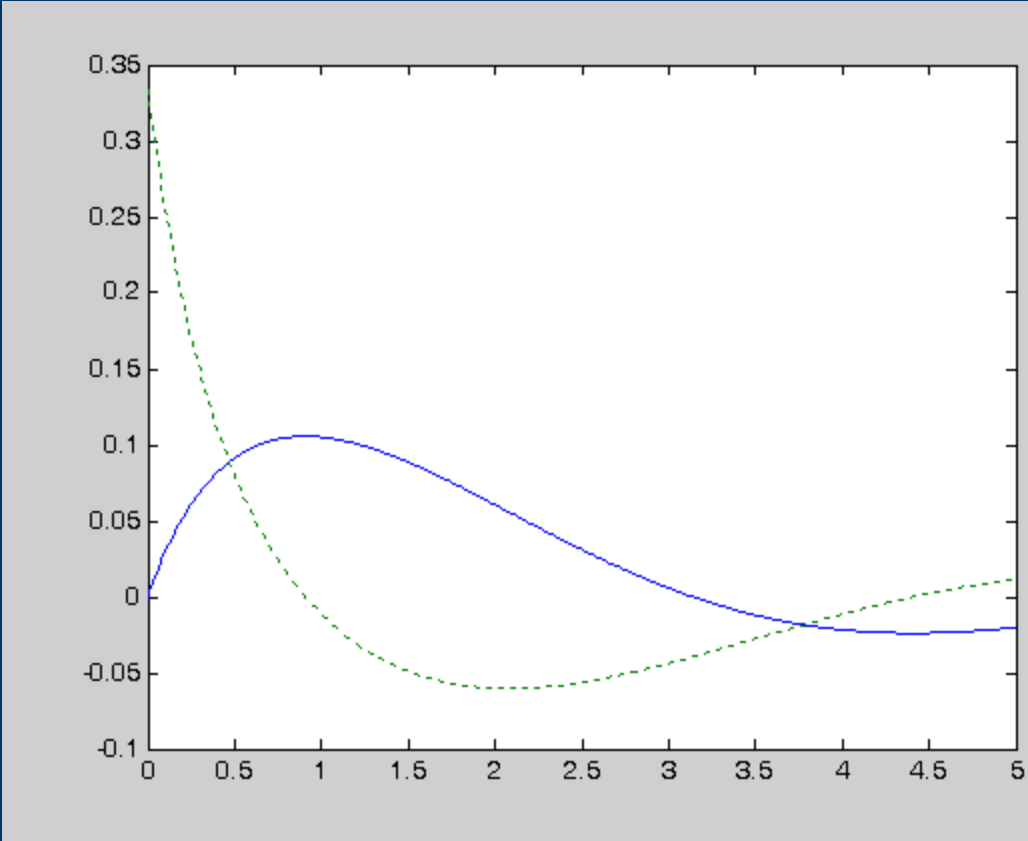
$$\frac{\cos(x)}{2x^2 + 4x + 3} - \frac{\sin(x)(2x + 4)}{(2x^2 + 4x + 3)^2}$$

原函数及一阶导数图：

```
>> x1=0:.01:5;
>> y=subs(f, x, x1);
>> y1=subs(f1, x, x1);
>> plot(x1,y,x1,y1,':')
```

更高阶导数：

```
>> tic, diff(f,x,100); toc
elapsed_time =
    4.6860
```



- 原函数4阶导数

```
>> f4=diff(f,x,4); pretty(f4)
```

$$\begin{aligned} & \frac{\sin(x)}{x^2 + 4x + 3} + 4 \frac{\cos(x) (2x + 4)}{(x^2 + 4x + 3)^2} - 12 \frac{\sin(x) (2x + 4)^2}{(x^2 + 4x + 3)^3} \\ & + 12 \frac{\sin(x)}{(x^2 + 4x + 3)^2} - 24 \frac{\cos(x) (2x + 4)^3}{(x^2 + 4x + 3)^4} + 48 \frac{\cos(x) (2x + 4)^3}{(x^2 + 4x + 3)^3} \\ & + 24 \frac{\sin(x) (2x + 4)^4}{(x^2 + 4x + 3)^5} - 72 \frac{\sin(x) (2x + 4)^2}{(x^2 + 4x + 3)^4} + 24 \frac{\sin(x)}{(x^2 + 4x + 3)^3} \end{aligned}$$

- 多元函数的偏导:

已知二元函数 $f(x, y)$, 求 $\frac{\partial^{m+n} f}{(\partial x^m \partial y^n)}$

- 格式: $f = \text{diff}(\text{diff}(f, x, m), y, n)$

或 $f = \text{diff}(\text{diff}(f, y, n), x, m)$

- 例: 已知 $z = f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$ 求其偏导数并用图表示。

$\frac{\partial z}{\partial x}$:

```
>> syms x y z=(x^2-2*x)*exp(-x^2-y^2-x*y);
```

```
>> zx=simple(diff(z,x))
```

$zx =$

$-\exp(-x^2 - y^2 - xy) * (-2*x + 2 + 2*x^3 + x^2*y - 4*x^2 - 2*x*y)$

$$\frac{\partial z}{\partial y}$$

```
>> zy=diff(z,y)
```

```
zy =
```

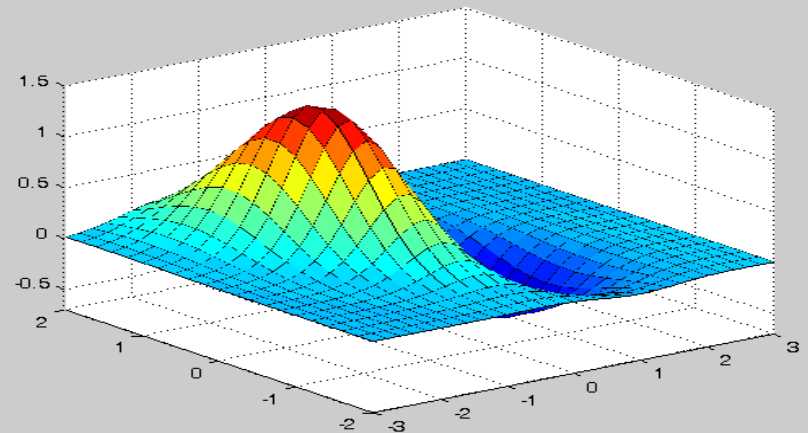
```
(x^2-2*x)*(-2*y-x)*exp(-x^2-y^2-x*y)
```

- 直接绘制三维曲面

```
>> [x,y]=meshgrid(-3:2:3,-2:2:2);
```

```
>> z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);
```

```
>> surf(x,y,z), axis([-3 3 -2 2 -0.7 1.5])
```

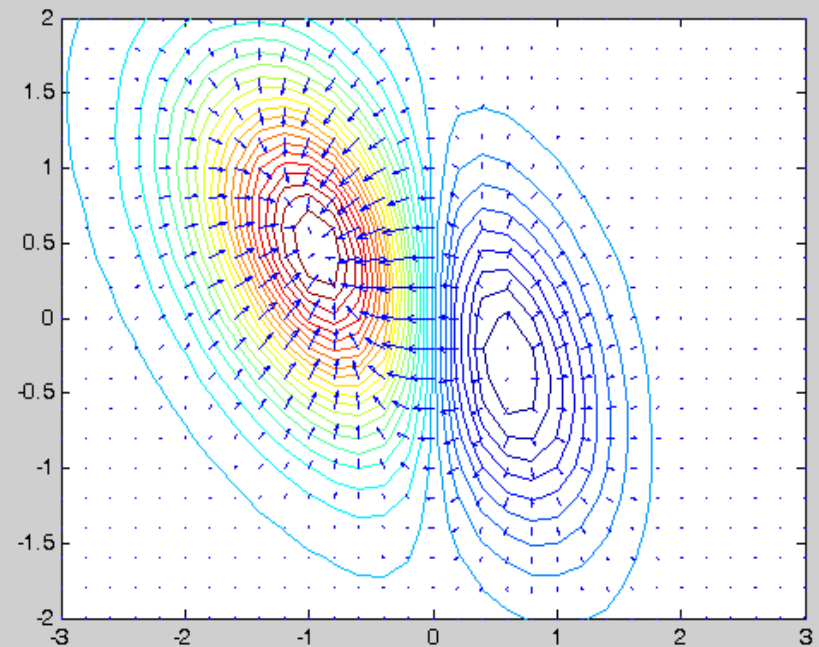
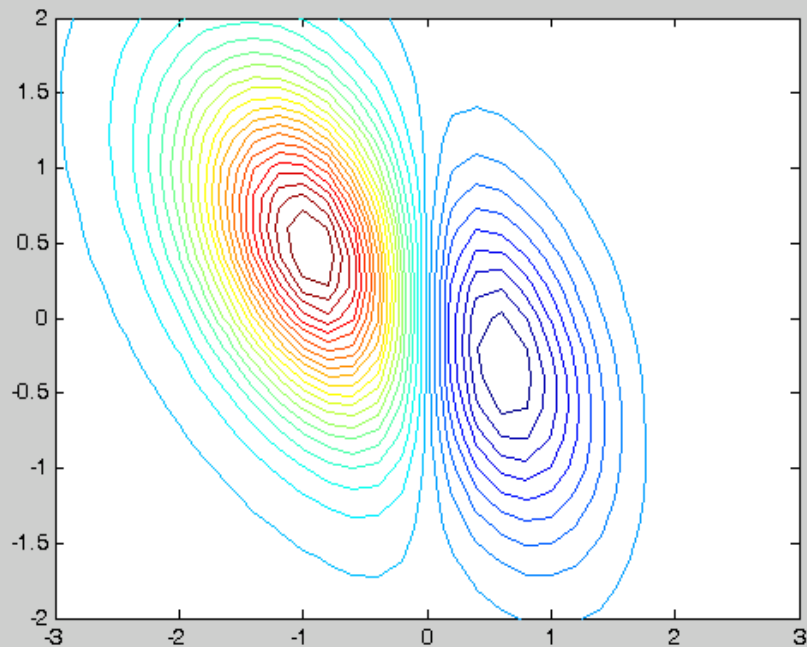


```
>> contour(x,y,z,30), hold on % 绘制等值线
```

```
>> zx=-exp(-x.^2-y.^2-x.*y).*(-2*x+2+2*x.^3+x.^2.*y-  
4*x.^2-2*x.*y);
```

```
>> zy=-x.*(x-2).*(2*y+x).*exp(-x.^2-y.^2-x.*y); % 偏导  
的数值解
```

```
>> quiver(x,y,zx,zy) % 绘制引力线
```



- 例 已知 $f(x, y, z) = \sin(x^2y)e^{-x^2y-z^2}$

$$\text{求 } \frac{\partial^4 f(x, y, z)}{(\partial x^2 \partial y \partial z)}$$

```
>> syms x y z; f=sin(x^2*y)*exp(-x^2*y-z^2);
>> df=diff(diff(diff(f,x,2),y),z); df=simple(df);
>> pretty(df)
```

$$\begin{aligned} & -4 z^2 \exp(-x^2 y - z^2) (\cos(x^2 y) - 10 \cos(x^2 y) y x^2 + 4 \\ & \sin(x^2 y) x^4 y + 4 \cos(x^2 y) x^2 y^2 - \sin(x^2 y)) \end{aligned}$$

6.2 数值微分

差商型求导公式

由导数定义 $f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$

(1) 向前差商公式

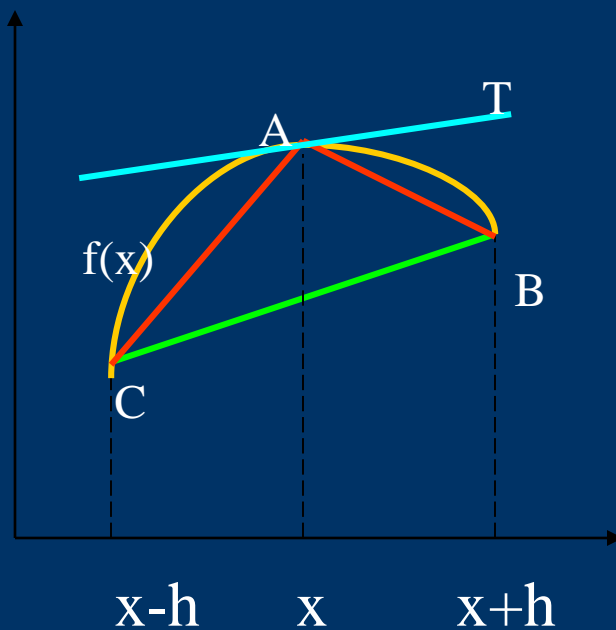
$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

(2) 向后差商公式

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

(3) 中心差商公式 (中点方法)

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$



6.2.1 数值微分算法

- 向前差商公式: $y'_i = \frac{\Delta y_i}{\Delta t} = \frac{y_{i+1} - y_i}{\Delta t}$
- 向后差商公式 $y'_i = \frac{\Delta y_i}{\Delta t} = \frac{y_i - y_{i-1}}{\Delta t}$

算法精度: $o(\Delta t)$

两种中心公式:

第一种:

$$y'_i = \frac{\Delta y_i}{\Delta t} = \frac{y_{i+1} - y_{i-1}}{2\Delta t}$$

$$\text{记为 } \widetilde{f}'(x) = \frac{f(x+\Delta t) - f(x-\Delta t)}{2\Delta t}$$

Taylor 级数展开:

$$\begin{aligned} \widetilde{f}'(x) &= \frac{f(x) + \Delta t f'(x) + \Delta t^2 f''(x)/2! + \Delta t^3 f'''(\xi)/3! + o(\Delta t^4)}{2\Delta t} \\ &\quad - \frac{f(x) - \Delta t f'(x) + \Delta t^2 f''(x)/2! - \Delta t^3 f'''(\xi)/3! + o(\Delta t^4)}{2\Delta t} \\ &= f'(x) + \frac{\Delta t^2}{3!} f'''(\xi) \end{aligned}$$

算法精度:

$$o(\Delta t^2)$$

高阶微分公式:

$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta t^2}$$

$$y_i''' = \frac{y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}}{2\Delta t^3}$$

$$y_i^{(4)} = \frac{y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}}{\Delta t^4}$$

第二种:

$$o(\Delta t^4)$$

微分公式:

$$y'_i = \frac{-y_{i+2} + 8y_{i+1} - 8y_{i-1} + y_{i-2}}{12\Delta t}$$

$$y''_i = \frac{-y_{i+2} + 16y_{i+1} - 30y_i + 16y_{i-1} - y_{i-2}}{12\Delta t^2}$$

$$y'''_i = \frac{-y_{i+3} + 8y_{i+2} - 13y_{i+1} + 13y_{i-1} - 8y_{i-2} + y_{i-3}}{8\Delta t^3}$$

$$y^{(4)}_i = \frac{-y_{i+3} + 12y_{i+2} - 39y_{i+1} + 56y_i - 39y_{i-1} + 12y_{i-2} - y_{i-3}}{6\Delta t^4}$$

6.2.2 中心差分方法及其 MATLAB 实现

```
function [dy,dx]=diff_ctr(y, Dt, n)
```

```
yx1=[y 0 0 0 0 0]; yx2=[0 y 0 0 0 0]; yx3=[0 0 y 0 0 0];
```

```
yx4=[0 0 0 y 0 0]; yx5=[0 0 0 0 y 0]; yx6=[0 0 0 0 0 y];
```

```
switch n
```

```
case 1
```

```
dy = (-diff(yx1)+7*diff(yx2)+7*diff(yx3)- ...  
diff(yx4))/(12*Dt); L0=3;
```

```
case 2
```

```
dy=(-diff(yx1)+15*diff(yx2)- 15*diff(yx3)...  
+diff(yx4))/(12*Dt^2);L0=3;
```

% 数值计算 `diff(X)` 表示数组 `X` 相邻两数的差

case 3

```
dy=(-diff(yx1)+7*diff(yx2)-6*diff(yx3)-6*diff(yx4)+...  
7*diff(yx5)-diff(yx6))/(8*Dt^3); L0=5;
```

case 4

```
dy = (-diff(yx1)+11*diff(yx2)-28*diff(yx3)+28*...  
diff(yx4)-11*diff(yx5)+diff(yx6))/(6*Dt^4); L0=5;  
end  
dy=dy(L0+1:end-L0); dx=([1:length(dy)]+L0-2-(n>2))*Dt;
```

调用格式: $[d_y, d_x] = \text{diff_ctr}(y, \Delta t, n)$

y为 等距实测数据, dy为得出的导数向量, dx为相应的自变量向量, dy、dx的数据比y短。

- 例: $f(x) = \sin x / (x^2 + 4x + 3)$

求导数的解析解, 再用数值微分求取原函数的1~4阶导数, 并和解析解比较精度。

```
>> h=0.05; x=0:h:pi;
```

```
>> syms x1; y=sin(x1)/(x1^2+4*x1+3);
```

```
% 求各阶导数的解析解与对照数据
```

```
>> yy1=diff(y); f1=subs(yy1,x1,x);
```

```
>> yy2=diff(yy1); f2=subs(yy2,x1,x);
```

```
>> yy3=diff(yy2); f3=subs(yy3,x1,x);
```

```
>> yy4=diff(yy3); f4=subs(yy4,x1,x);
```

```

>> y=sin(x)./(x.^2+4*x+3); % 生成已知数据点
>> [y1,dx1]=diff_ctr(y,h,1); subplot(221),plot(x,f1,dx1,y1,':');
>> [y2,dx2]=diff_ctr(y,h,2); subplot(222),plot(x,f2,dx2,y2,':');
>> [y3,dx3]=diff_ctr(y,h,3); subplot(223),plot(x,f3,dx3,y3,':');
>> [y4,dx4]=diff_ctr(y,h,4); subplot(224),plot(x,f4,dx4,y4,':')

```

求最大相对误差:

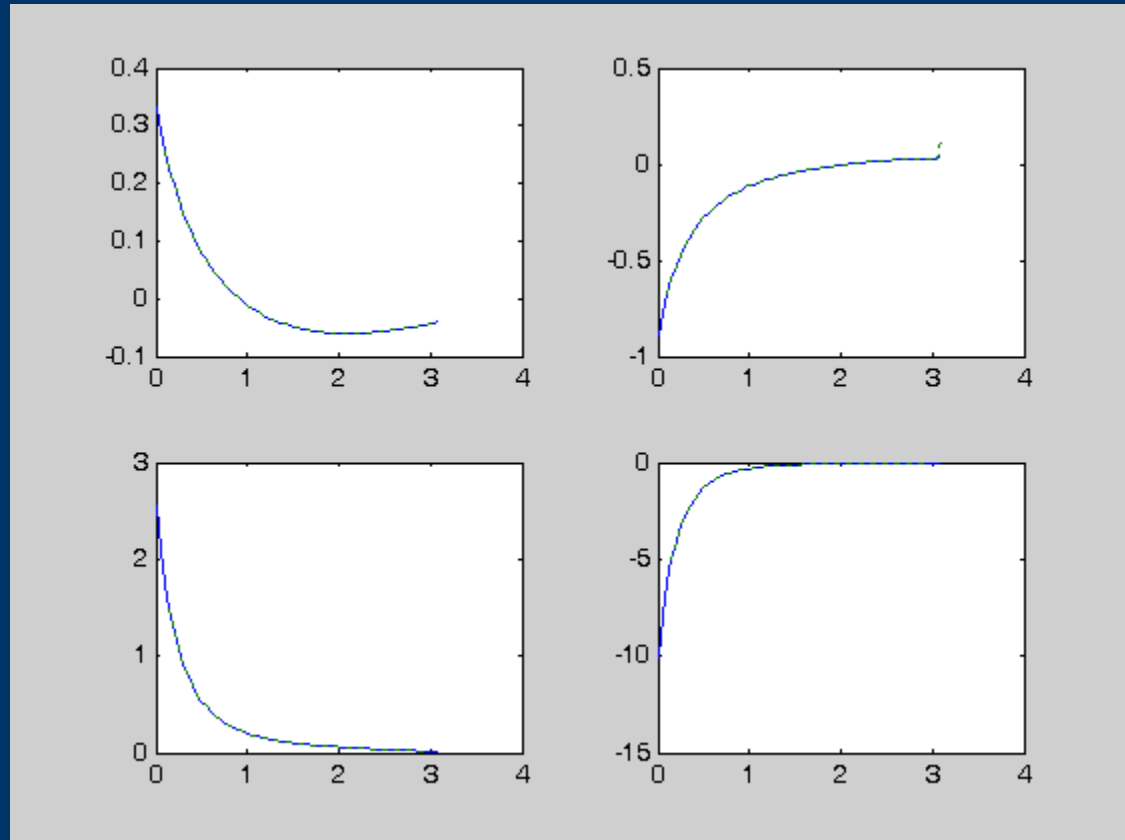
```

>> norm((y4-...
f4(4:60))./f4(4:60))

```

ans =

3.5025e-004



6.2.3 用插值、拟合多项式的求导数

- 基本思想：当已知函数在一些离散点上的函数数值时，该函数可用插值或拟合多项式来近似，然后对多项式进行微分求得导数。
- 选取 $x=0$ 附近的少量点

$$(x_i, y_i), \quad i = 1, 2, \dots, n+1$$

- 进行多项式拟合或插值

$$g(x) = c_1 x^n + c_2 x^{n-1} + \dots + c_n x + c_{n+1}$$

- $g(x)$ 在 $x=0$ 处的 k 阶导数为

$$g^{(k)}(0) = c_{n+1-k} k! \quad k = 0, 1, 2, \dots, n$$

- 通过坐标变换用上述方法计算任意x点处的导数值
- 令 $z = x - a$
- 将g(x)写成z的表达式

$$g(x) = \bar{g}(z) = d_1 z^n + d_2 z^{n-1} + \cdots + d_n z + d_{n+1}$$

- 导数为

$$g^{(k)}(a) = \bar{g}^{(k)}(0) = d_{n+1-k} k! \quad k = 0, 1, \cdots, n$$

- 可直接用 $\bar{g}(z)$ 拟合节点 $(x_i - a, y_i)$ 得到系数 d_i

$$\mathbf{d} = \text{polyfit}(\mathbf{x} - \mathbf{a}, \mathbf{y}, \text{length}(\mathbf{x}) - 1)$$

- 例：数据集集合如下：

xd: 0 0.2000 0.4000 0.6000 0.8000 1.000

yd: 0.3927 0.5672 0.6982 0.7941 0.8614 0.9053

计算 $x=a=0.3$ 处的各阶导数。

```
>> xd=[ 0 0.2000 0.4000 0.6000 0.8000 1.000];
```

```
>> yd=[0.3927 0.5672 0.6982 0.7941 0.8614 0.9053];
```

```
>> a=0.3;L=length(xd);
```

```
>> d=polyfit(xd-a,yd,L-1);fact=[1];
```

```
>> for k=1:L-1;fact=[factorial(k),fact];end
```

```
>> deriv=d.*fact
```

```
deriv =
```

```
1.8750 -1.3750 1.0406 -0.9710 0.6533 0.6376
```


- 建立用拟合（插值）多项式计算各阶导数的
poly_drv.m

```
function der=poly_drv(xd,yd,a)
```

```
m=length(xd)-1;
```

```
d=polyfit(xd-a,yd,m);
```

```
c=d(m:-1:1); %去掉常数项
```

```
fact(1)=1;for i=2:m; fact(i)=i*fact(i-1);end
```

```
der=c.*fact;
```

- 例：

```
>> xd=[ 0 0.2000 0.4000 0.6000 0.8000 1.000];
```

```
>> yd=[0.3927 0.5672 0.6982 0.7941 0.8614 0.9053];
```

```
>> a=0.3; der=poly_drv(xd,yd,a)
```

```
der =
```

```
0.6533 -0.9710 1.0406 -1.3750 1.8750
```

6.2.4 二元函数的梯度计算

二元函数的函数值矩阵 z ，其中 z 为网格数据

$$z = f(x, y)$$

- 格式:

$$[f_x, f_y] = \text{gradient}(z)$$

- 若 z 矩阵是建立在等间距的形式生成的网格基础上，则实际梯度为

$$f_x = f_x / \Delta x, \quad f_y = f_y / \Delta y$$

Δx 和 Δy 分别为 x, y 生成网格的步距

• 例： 已知 $z = f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$

计算梯度， 绘制引力线图：

```
>> [x,y]=meshgrid(-3:.2:3,-2:.2:2); z=(x.^2-2*x).*exp(-  
x.^2-y.^2-x.*y);
```

```
>> [fx,fy]=gradient(z);
```

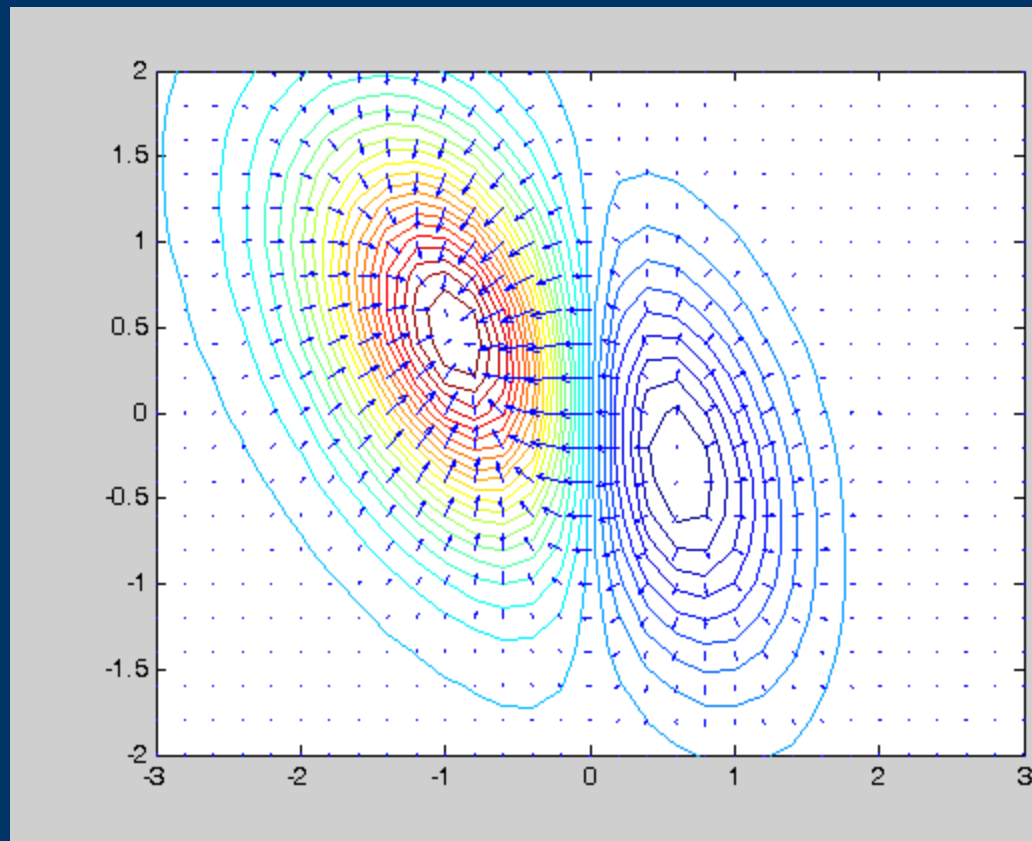
```
>> fx=fx/0.2; fy=fy/0.2;
```

```
>> contour(x,y,z,30);
```

```
>> hold on;
```

```
>> quiver(x,y,fx,fy)
```

% 绘制等高线与
引力线图



- 绘制误差曲面:

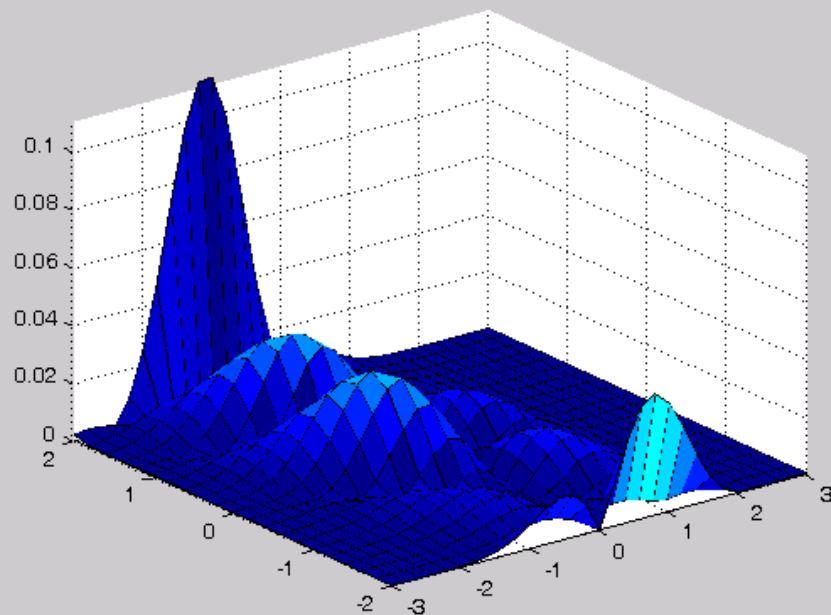
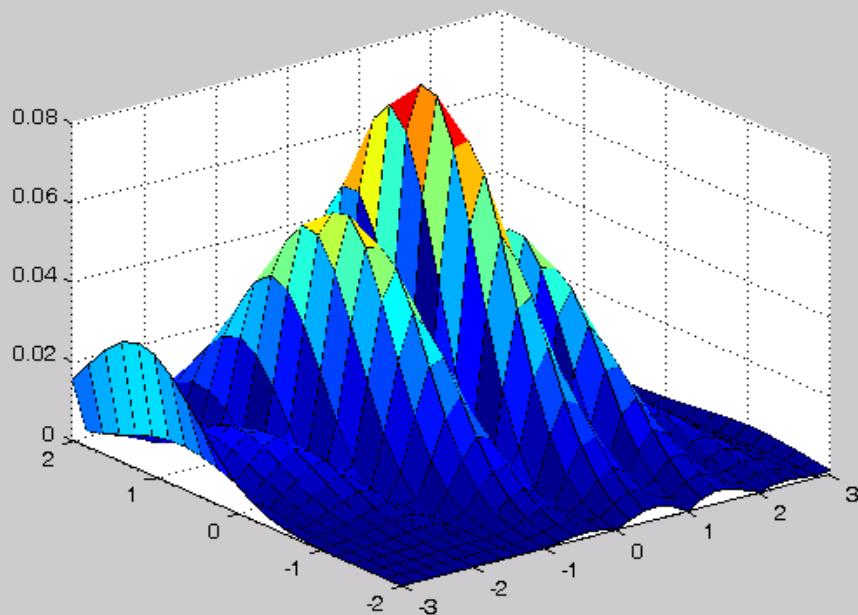
```
>> zx=-exp(-x.^2-y.^2-x.*y).*(-2*x+2+2*x.^3+x.^2.*y-4*x.^2-  
2*x.*y);
```

```
>> zy=-x.*(x-2).*(2*y+x).*exp(-x.^2-y.^2-x.*y);
```

```
>> surf(x,y,abs(fx-zx)); axis([-3 3 -2 2 0,0.08])
```

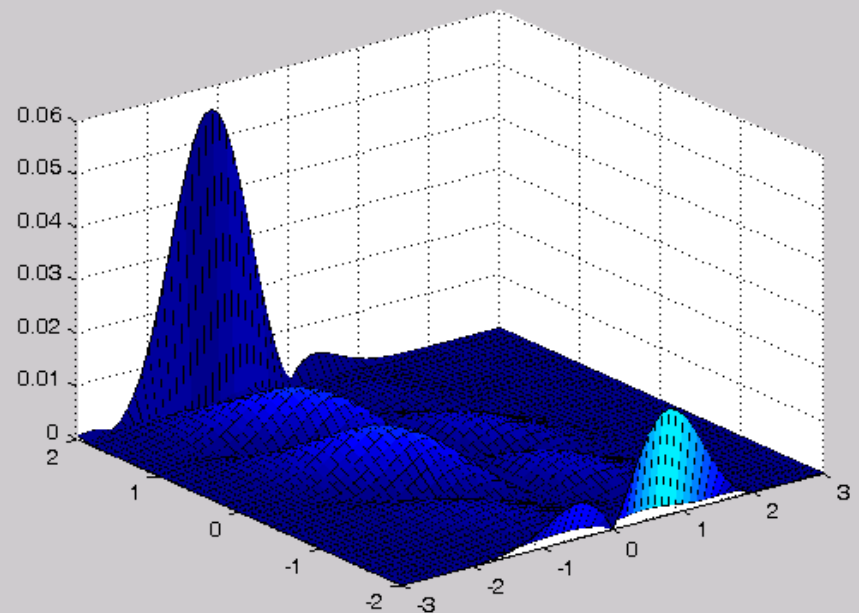
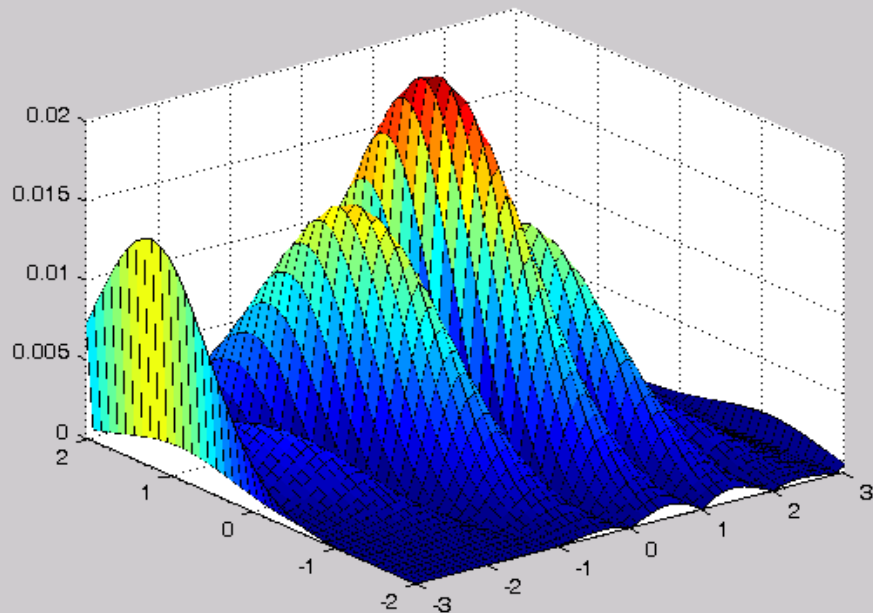
```
>> figure; surf(x,y,abs(fy-zy)); axis([-3 3 -2 2 0,0.11])
```

%建立一个新图形窗口



- 为减少误差，对网格加密一倍：

```
>> [x,y]=meshgrid(-3:1:3,-2:1:2); z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);  
>> [fx,fy]=gradient(z); fx=fx/0.1; fy=fy/0.1;  
>> zx=-exp(-x.^2-y.^2-x.*y).*(-2*x+2+2*x.^3+x.^2.*y-4*x.^2-2*x.*y);  
>> zy=-x.*(x-2).*(2*y+x).*exp(-x.^2-y.^2-x.*y);  
>> surf(x,y,abs(fx-zx)); axis([-3 3 -2 2 0,0.02])  
>> figure; surf(x,y,abs(fy-zy)); axis([-3 3 -2 2 0,0.06])
```



6.3 函数的级数展开与 级数求和问题求解

- 6.3.1 Taylor 幂级数展开
- 6.3.2 Fourier 级数展开
- 6.3.3 级数求和的计算

6.3.1 Taylor 幂级数展开

6.3.1.1 单变量函数的 Taylor 幂级数展开

在 $x = 0$ 点附近的 Taylor 幂级数:

$$f(x) = a_1 + a_2x + a_3x^2 + \cdots + a_kx^{k-1} + o(x^k)$$

$$\text{其中 } a_i = \frac{1}{i!} \lim_{x \rightarrow 0} \frac{d^{i-1}}{dx^{i-1}} f(x), \quad i = 1, 2, 3, \cdots$$

关于 $x = a$ 点展开:

$$f(x) = b_1 + b_2(x-a) + b_3(x-a)^2 + \cdots + b_k(x-a)^{k-1} + o[(x-a)^k]$$

$$\text{其中 } b_i = \frac{1}{i!} \lim_{x \rightarrow a} \frac{d^{i-1}}{dx^{i-1}} f(x), \quad i = 1, 2, 3, \dots$$

```
taylor(f, x, k)
```

```
% 按  $x = 0$  进行 Taylor 幂级数展开
```

```
taylor(f, x, k, a)
```

```
% 按  $x = a$  进行 Taylor 幂级数展开
```


例: $f(x) = \sin x / (x^2 + 4x + 3)$

求其 Taylor 幂级数展开的前 9 项,

并关于 $x = 2$ 和 $x = a$ 进行 Taylor 幂级数展开。

```
>> syms x; f=sin(x)/(x^2+4*x+3);
```

```
>> y1=taylor(f,x,9); pretty(y1)
```

```
      2  23 3   34 4   4087 5   3067 6   515273 7   386459 8
1/3 x - 4/9 x + -- x - ---- x + -----x - ----- x +----- x - ----- x
      54      81      9720    7290    1224720    918540
```

```
>> taylor(f,x,9,2)
```

```
ans =
```

```
1/15*sin(2)+(1/15*cos(2)-8/225*sin(2))*(x-2)+ (-127/6750*sin(2)-  
8/225*cos(2))*(x-2)^2 +(23/6750*cos(2)+628/50625*sin(2))*(x-2)^3 +(-  
15697/6075000*sin(2)+28/50625*cos(2))*(x-2)^4  
+(203/6075000*cos(2)+6277/11390625*sin(2))*(x-2)^5 +(-  
585671/2733750000*sin(2)-623/11390625*cos(2))*(x-2)^6  
+(262453/19136250000*cos(2)+397361/5125781250*sin(2))*(x-2)^7 +(-  
875225059/344452500000000*sin(2)-131623/35880468750*cos(2))*(x-2)^8
```

```
>> syms a; taylor(f,x,5,a) % 结果较冗长，显示从略
```

```
ans =
```

```
sin(a)/(a^2+3+4*a) +(cos(a)-  
sin(a)/(a^2+3+4*a)*(4+2*a))/(a^2+3+4*a)*(x-a) +(-  
sin(a)/(a^2+3+4*a)-1/2*sin(a)-(cos(a)*a^2+3*cos(a)+4*cos(a)*a-  
4*sin(a)-2*sin(a)*a)/(a^2+3+4*a)^2*(4+2*a))/(a^2+3+4*a)*(x-  
a)^2+...
```

例:对 $y=\sin x$ 进行Taylor幂级数展开,并观察不同阶次的近似效果。

```
>> x0=-2*pi:0.01:2*pi; y0=sin(x0); syms x; y=sin(x);  
>> plot(x0,y0,'r-.'), axis([-2*pi,2*pi,-1.5,1.5]); hold on  
>> for n=[8:2:16]
```

```
    p=taylor(y,x,n), y1=subs(p,x,x0); line(x0,y1) end
```

p =

$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7$

p =

$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9$

p =

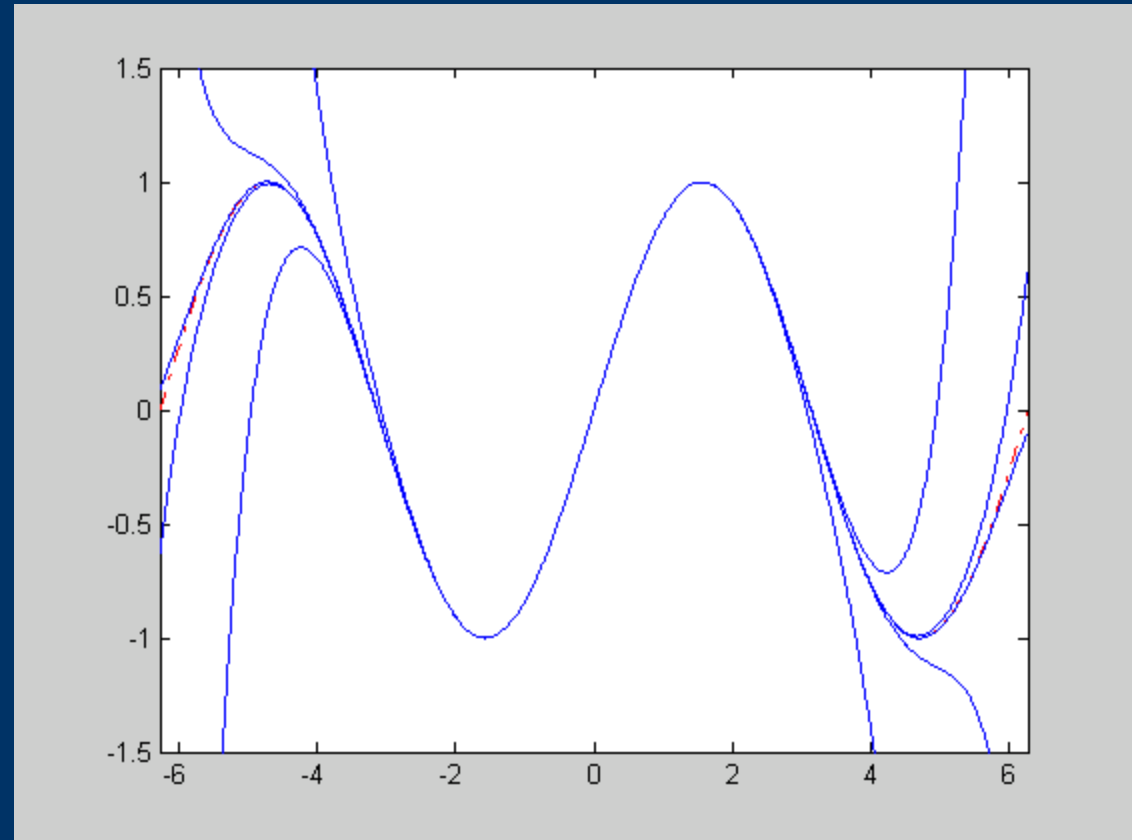
$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 - \frac{1}{39916800}x^{11}$

p =

$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 - \frac{1}{39916800}x^{11} + \frac{1}{6227020800}x^{13}$

$p =$

$$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 - \frac{1}{39916800}x^{11} + \frac{1}{6227020800}x^{13} - \frac{1}{1307674368000}x^{15}$$



6.3.1.2 多变量函数的Taylor 幂级数展开

- 多变量函数 $f(x_1, x_2, \dots, x_n)$ 在 (a_1, a_2, \dots, a_n) 的Taylor幂级数的展开

$$\begin{aligned} f(x_1, \dots, x_n) = & f(a_1, \dots, a_n) + \\ & \left[(x_1 - a_1) \frac{\partial}{\partial x_1} + \dots + (x_n - a_n) \frac{\partial}{\partial x_n} \right] f(a_1, \dots, a_n) + \\ & \frac{1}{2!} \left[(x_1 - a_1) \frac{\partial}{\partial x_1} + \dots + (x_n - a_n) \frac{\partial}{\partial x_n} \right]^2 f(a_1, \dots, a_n) + \dots + \\ & \frac{1}{k!} \left[(x_1 - a_1) \frac{\partial}{\partial x_1} + \dots + (x_n - a_n) \frac{\partial}{\partial x_n} \right]^k f(a_1, \dots, a_n) + \dots \end{aligned}$$

```
F=maple('mtaylor',f,'[x1,...,xn]',k)
```

根据原点展开

```
F=maple('mtaylor',f,'[x1 = a1,...,xn = an]',k)
```

根据 (a_1, \dots, a_n) 点展开

- 例：求 $z = f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$ 的各种 Taylor 幂级数展开。

```
>> syms x y; f=(x^2-2*x)*exp(-x^2-y^2-x*y);
```

```
>> F=maple('mtaylor',f,'[x,y]',8)
```

F =

```
mtaylor((x^2-2*x)*exp(-x^2-y^2-x*y),[x, y],8)
```

```
>> maple('readlib(mtaylor)'); % 读库，把函数调入内存
```

```
>> F=maple('mtaylor',f,'[x,y]',8)
```

F =

```
-2*x+x^2+2*x^3-x^4-x^5+1/2*x^6+1/3*x^7+2*y*x^2+2*y^2*x-  
y*x^3-y^2*x^2-2*y*x^4-3*y^2*x^3-2*y^3*x^2-  
y^4*x+y*x^5+3/2*y^2*x^4+y^3*x^3+1/2*y^4*x^2+y*x^6+2*y^  
2*x^5+7/3*y^3*x^4+2*y^4*x^3+y^5*x^2+1/3*y^6*x
```

```
>> syms a; F=maple('mtaylor',f,'[x=1,y=a]',3);
```

```
>> F=maple('mtaylor',f,'[x=a]',3)
```

F =

```
(a^2-2*a)*exp(-a^2-y^2-a*y)+((a^2-2*a)*exp(-a^2-y^2-a*y))*(-2*a-  
y)+(2*a-2)*exp(-a^2-y^2-a*y))*(x-a)+((a^2-2*a)*exp(-a^2-y^2-  
a*y))*(-1+2*a^2+2*a*y+1/2*y^2)+exp(-a^2-y^2-a*y)+(2*a-  
2)*exp(-a^2-y^2-a*y))*(-2*a-y))*(x-a)^2
```

6.3.2 Fourier 级数展开

给定 $f(x)$, $x \in [-L, L]$, $T = 2L$, $f(t) = f(kT + t)$

Fourier 级数:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi}{L}x + b_n \sin \frac{n\pi}{L}x \right)$$

$$\text{其中} \begin{cases} a_n = \frac{1}{L} \int_{-L}^L f(x) \cos \frac{n\pi x}{L} dx, & n = 0, 1, 2, \dots \\ b_n = \frac{1}{L} \int_{-L}^L f(x) \sin \frac{n\pi x}{L} dx, & n = 1, 2, 3, \dots \end{cases}$$

Fourier 级数的解析函数：

```
function [A,B,F]=fseries(f,x,n,a,b)
if nargin==3, a=-pi; b=pi; end
L=(b-a)/2;
if a+b, f=subs(f,x,x-L-a); end % 变量区域互换
A=int(f,x,-L,L)/L; B=[]; F=A/2; % 计算a0
for i=1:n
    an=int(f*cos(i*pi*x/L),x,-L,L)/L;
    bn=int(f*sin(i*pi*x/L),x,-L,L)/L; A=[A, an]; B=[B,bn];
    F=F+an*cos(i*pi*x/L)+bn*sin(i*pi*x/L);
end
if a+b, F=subs(F,x,x+L+a); end % 换回变量区域
```

fseries() 函数:

$$[A, B, F] = \text{fseries}(f, x, n, a, b)$$

例: 求 $y = x(x - \pi)(x - 2\pi)$, $x \in (0, 2\pi)$
的 Fourier 级数展开。

```
>> syms x; f=x*(x-pi)*(x-2*pi);
```

```
>> [A,B,F]=fseries(f,x,6,0,2*pi)
```

A =

[0, 0, 0, 0, 0, 0, 0]

B =

[-12, 3/2, -4/9, 3/16, -12/125, 1/18]

F =

$12*\sin(x)+3/2*\sin(2*x)+4/9*\sin(3*x)+3/16*\sin(4*x)+12/125*\sin(5*x)+1/18*\sin(6*x)$

例: $(-\pi, \pi)$ 区间的方波信号, 设 $x \geq 0$ 时 $y = 1$, 否则 $y = -1$, 对它进行 Fourier 级数拟合, 并观察用多少项能有较好的拟合效果?

```
>> syms x; f=abs(x)/x; % 定义方波信号
>> xx=[-pi:pi/200:pi]; xx=xx(xx~=0); xx=sort([xx,-eps,eps]); % 剔除零点
>> yy=subs(f,x,xx); plot(xx,yy,'r-.'), hold on % 绘制出理论值并保持坐标系
>> for n=2:20
    [a,b,f1]=fseries(f,x,n), y1=subs(f1,x,xx);
    plot(xx,y1)
end
```

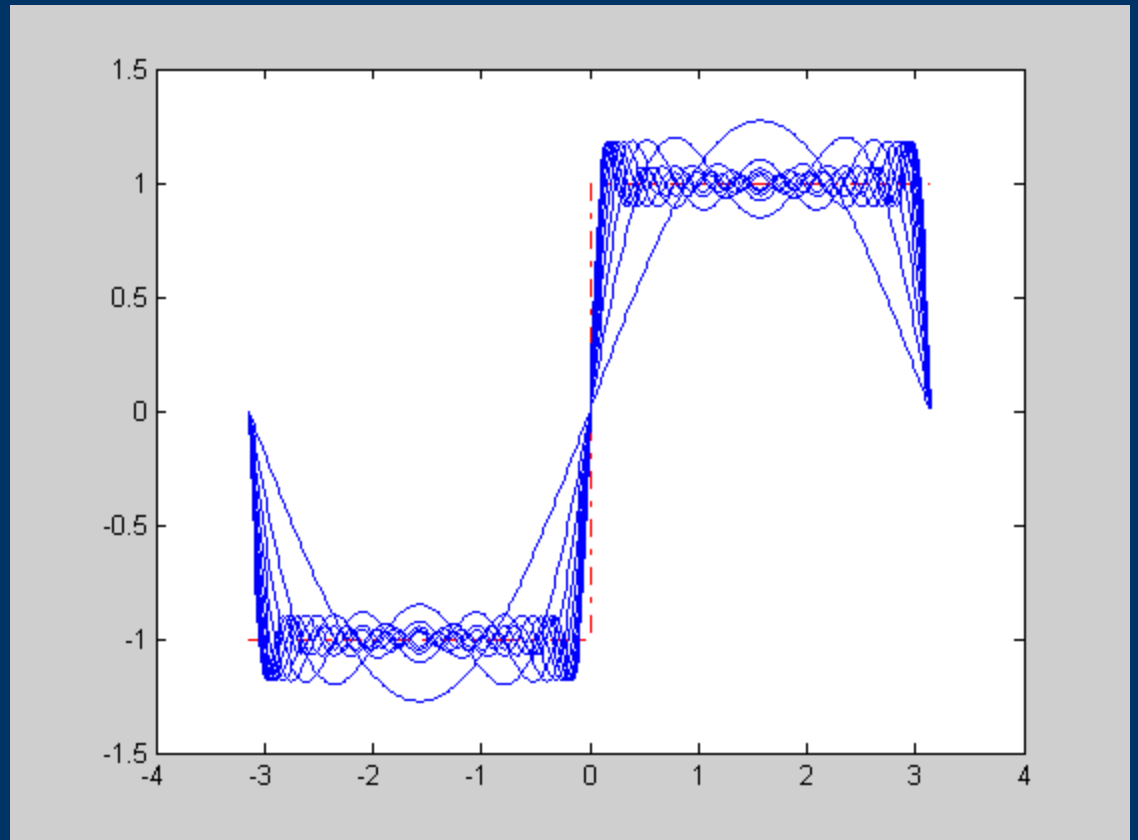
a =
[0, 0, 0]
b =
[4/pi, 0]
f1 =
4/pi*sin(x)

a =
[0, 0, 0, 0]
b =

[4/pi, 0, 4/3/pi]

f1 =
4/pi*sin(x)+4/3/pi*sin(3*x)

.....



6.3.3 级数求和的计算

- 是在符号工具箱中提供的

$$\text{求 } S = \sum_{k=k_0}^{k_n} f_k$$

$$S = \text{symsum}(f_k, k, k_0, k_n)$$

例:计算

$$S = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + \cdots + 2^{62} + 2^{63} = \sum_{i=0}^{63} 2^i$$

```
>> format long; sum(2.^[0:63]) % 数值计算
```

```
ans =
```

```
1.844674407370955e+019
```

```
>> sum(sym(2).^[0:200]) % 或 syms k; symsum(2^k,0,200)  
% 把2定义为符号量可使计算更精确
```

```
ans =
```

```
32138760885179805510839241846823252050444059875655856706  
02751
```

```
>> syms k; symsum(2^k,0,200)
```

```
ans =
```

```
32138760885179805510839241846823252050444059875655856706  
02751
```

例:试求解无穷级数的和

$$S = \frac{1}{1 \times 4} + \frac{1}{4 \times 7} + \frac{1}{7 \times 10} + \cdots + \frac{1}{(3n-2)(3n+1)} + \cdots$$

```
>> syms n; s=symsum(1/((3*n-2)*(3*n+1)),n,1,inf)
```

%采用符号运算工具箱

```
s =
```

```
1/3
```

```
>> m=1:10000000; s1=sum(1./((3*m-2).*(3*m+1)));% 数值计算方法,双精度有效位16,“大数吃小数”,无法精确
```

```
>> format long; s1 % 以长型方式显示得出的结果
```

```
s1 =
```

```
0.333333332222165
```

例:求解

$$J = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)(2x+1)^{2n+1}}$$

```
>> syms n x
```

```
>> s1=symsum(2/((2*n+1)*(2*x+1)^(2*n+1)),n,  
0,inf);
```

```
>> simple(s1) % 对结果进行化简, MATLAB  
6.5 及以前版本因本身 bug 化简很麻烦
```

```
ans =
```

```
log((((2*x+1)^2)^(1/2)+1)/(((2*x+1)^2)^(1/2)-1))
```

```
% 实际应为log((x+1)/x)
```


例:求 $\lim_{n \rightarrow \infty} \left[\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} \right) - \ln n \right]$

```
>> syms m n; limit(symsum(1/m,m,1,n)-log(n),n,inf)
```

```
ans =
```

```
eulergamma
```

```
>> vpa(ans, 70) % 显示 70 位有效数字
```

```
ans =
```

```
.577215664901532860606512090082402431042159  
3359399235988057672348848677
```