# 第6章　MATLAB符号计算

在MATLAB中，提供了强大的符号运算功能，并且有专门的符号数学工具箱Symbolic Math Toolbox。此外，在MATLAB中，还可以通过maple.m和map.m两个接口和Maple相连。

# 第6章　MATLAB符号计算

**在MATLAB中，符号计算函数主要分为以下几类：**

- 符号表达式
- 符号矩阵操作
- 符号微积分
- 符号积分变换
- 符号方程求解
- 符号函数的绘图
- 图形化符号函数计算器

# 6.1　符号运算入门

- 如何生成符号型数据变量
- 利用符号变量产生函数表达式和方程
- 符号变量的基本运算

## 6.1.1　符号变量的创建

符号数学工具箱中定义了一种新的数据类型：**sym**类，**sym**类的实例就是符号对象，用来存储代表符号的字符串。

在**MATLAB**中，提供了两个建立符号对象的函数：函数**sym( )**和函数**syms( )**，两个函数的用法不同。

## 函数sym()：用来建立单个符号量

S=sym(A)：该函数由输入参数A建立符号对象S，输出参数的类型为sym。输入参数A不带单引号，表示是一个由数字、数值矩阵或表达式转换成的符号矩阵。

S=sym(A, flag)：输入参数flag为转换的符号对象应该符合的格式类型。

S=sym('A')：输入参数A带单引号，表示A是一个字符串，输出是由字符串转换成的符号对象。符号字符串可以是常量、变量、函数或表达式。

## 函数syms()：用来一次建立多个符号变量

syms(S1 S2 S3)：该函数一次定义3个符号对象，符号对象之间用空格进行分隔，不能用逗号分隔。

syms(S1 S2 S3 flag)：该函数的输入参数flag为转换的符号对象应该符合的格式。

## 函数class( )：获得输入对象的数据类型

class(obj)：其中输入参数obj为要判断的对象，返回该对象的类型C。如果输入为符号对象，则返回类型为sym。

# 例：利用函数sym建立符号常量

| | |
|---|---|
| a1=sqrt(5) | a1 = 2.2361 |
| a2=sqrt(sym(5)) | a2 = 5^(1/2) |
| a3=double(a2) | a3 = 2.2361 |
| a4=sym(3)/sym(8) | a4 = 3/8 |
| a5=3*a4 | a5 = 9/8 |
| a6='3/8' | a6 = 3/8 |
| ca1=class(a1) | ca1 = double |
| ca2=class(a2) | ca2 = sym |
| ca3=class(a3) | ca3 = double |
| ca4=class(a4) | ca4 = sym |
| ca5=class(a5) | ca5 = sym |
| ca6=class(a6) | ca6 = char |

## 在matlab命令窗口输入whos后，显示如下：

| Name | Size | Bytes | Class | Attributes |
|---|---|---|---|---|
| a1 | 1x1 | 8 | double | |
| a2 | 1x1 | 60 | sym | |
| a3 | 1x1 | 8 | double | |
| a4 | 1x1 | 60 | sym | |
| a5 | 1x1 | 60 | sym | |
| a6 | 1x3 | 6 | char | |
| ca1 | 1x6 | 12 | char | |
| ca2 | 1x3 | 6 | char | |
| ca3 | 1x6 | 12 | char | |
| ca4 | 1x3 | 6 | char | |
| ca5 | 1x3 | 6 | char | |
| ca6 | 1x4 | 8 | char | |

# 例：利用函数sym和syms建立符号变量

```
x=sym('x');
y=sym('y');
syms a b;
classx=class(x)
classy=class(y)
classa=class(a)
classb=class(b)
```

```
classx =
        sym
classy =
        sym
classa =
        sym
classb =
        sym
```

## 例：利用函数sym建立符号表达式$ax^2+bx+c$

```
a=sym('a');
b=sym('b');
c=sym('c');
x=sym('x');
f1=a*x^2+b*x+c
g=f1^2+3*f1+5
f2=sym('a*x^2+b*x+c')
```

```
f1 =    a*x^2 + b*x + c
g =     3*c + 3*b*x + 3*a*x^2 + (a*x^2 + b*x + c)^2 + 5
f2 =    a*x^2 + b*x + c
```

例：利用函数syms建立符号表达式$ax^2+bx+c$

```
syms a b c x;
f1=a*x^2+b*x+c
g=f1^2+3*f1+5
```

```
f1 =    a*x^2 + b*x + c
g =     3*c + 3*b*x + 3*a*x^2 + (a*x^2 + b*x + c)^2 + 5
```

## 6.1.2　符号函数和符号方程

在MATLAB中，符号表达式是由符号常量、符号变量、符号函数运算符以及专用函数连接起来的符号对象。

符号表达式有两类：符号函数和符号方程。方程与函数的区别为：函数是一个代数式，而方程是一个等式。

例：建立符号函数$f_1=4(x-2)^2+\sin(x+y)+5^x$和$f_2=f_1/7x$

```
syms x y;
f1=4*(x-2)^2+sin(x+y)+5^x
f2=f1/(7*x)
f3=sym(f1/(7*x))
```

```
f1 =
        sin(x + y) + 4*(x - 2)^2 + 5^x
f2 =
        (sin(x + y) + 4*(x - 2)^2 + 5^x)/(7*x)
f3 =
        (sin(x + y) + 4*(x - 2)^2 + 5^x)/(7*x)
```

例：建立符号方程$4(x-2)^2+5^y=0$和$5x^2+4x+2=0$

```
syms x y;
e1=sym('4*(x-2)^2+5^y=0')
e2=sym('5*x^2+4*x+2=0')
```

```
e1 =
        4*(x - 2)^2 + 5^y = 0
e2 =
        5*x^2 + 4*x + 2 = 0
```

# 6.2　简单实例分析

## 求解一元二次方程的根

对于一元二次方程，$ax^2+bx+c=0$，其中$a\neq0$，根据一元二次方程的求根公式，得到方程的两个根。当$b^2-4ac\geq0$时，该方程有实根；当$b^2-4ac<0$时，方程有两个共轭的复根。

```
sym x
y1=solve('a*x^2+b*x+c=0')
y2=solve('3*x^2+8*x+5=0')
y3=solve('x^2-2*x+6=0')
```

```
y1 =
        -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
        -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
y2 =
         -5/3
           -1
y3 =
         1 - 5^(1/2)*i
         1 + 5^(1/2)*i
```

# 6.3　符号运算精度

在MATLAB中，采用函数digits( )和vpa( )来实现任意精度的符号运算。下面分别进行介绍：

digits(d)：调用该函数后，符号对象的近似解的精度变成d位有效数字，参数d的默认值为32位。

D=digits：调用该函数后，得到当前采用的数值计算的精度。

R=vpa(A)：该函数计算符号矩阵A的近似解，精度为函数digits(d)指定的有效位数。

R=vpa(A, d)：该函数计算符号矩阵A的近似解，有效位数由参数d决定。

```
digits
a1=sym(1.3,'d')
digits(40)
digits
a2=sym(1.3,'d')
```

```
Digits = 32
a1 =
        1.300000000000000044408920985006
Digits = 40
a2 =
        1.300000000000000044408920985006261616945
```

按照整数位置的二分法来取精度

```
a=vpa(hilb(2))
b=vpa(hilb(3),6)
c=vpa(pi)
d=vpa(pi,100)
```

```
a =     [ 1.0,                              0.5]
        [ 0.5, 0.33333333333333333333333333333333]
b =     [      1.0,      0.5, 0.333333]
        [      0.5, 0.333333,     0.25]
        [ 0.333333,     0.25,      0.2]
c =     3.1415926535897932384626433832795
d =
3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342117068
```

# 6.4　符号表达式的操作

符号表达式可以进行加减乘除的四则运算，此外，还可以对符号表达式进行因式分解、展开、合并同类项、获取分子和分母等，这些操作都非常的简单方便，下面分别进行介绍。

# 6.4.1　符号表达式的基本运算

在MATLAB中，采用了函数重载技术，使得符号表达式的运算符和基本函数与数值计算中的运算符和基本函数几乎完全相同。符号表达式可以进行“+”、“-”、“*”、“/”四则运算。

符号表达式的比较中，只有运算符“==”和“~=”代表“等于”和“不等于”。当结果为“真”时，返回值为1，否则返回值为0。

```
syms x y z;
f1=sin(x)+acos(y)
f2=sqrt(x)+exp(y)
f3=log2(x)+log(y)
g1=f1+f2
g2=f1*f2
g3=f1/f2
f1==f2
```

```
f1 =    acos(y) + sin(x)
f2 =    exp(y) + x^(1/2)
f3 =    log(y) + log(x)/log(2)
g1 =    acos(y) + exp(y) + sin(x) + x^(1/2)
g2 =    (exp(y) + x^(1/2))*(acos(y) + sin(x))
g3 =    (acos(y) + sin(x))/(exp(y) + x^(1/2))
ans =    0
```

## 6.4.2　符号表达式的常用操作

符号表达式的常用操作，包括：

- 函数findsym( )寻找符号变量。
- 函数factor( )进行符号多项式的因式分解。
- 函数expand( )进行符号表达式的展开。
- 函数collect( )进行符号表达式中同类项的合并。
- 函数horner( )将符号多项式转换成嵌套形式。
- 函数numden( )获取符号表达式的分子和分母。

## 函数findsym( )：寻找符号变量

**findsym(S)**：寻找函数符号表达式**S**中所有的符号变量；

**findsym(S, n)**：寻找函数符号表达式**S**中**n**个在字母表中与**x**最接近的符号变量；

```
syms m n x y;
f=m^2+5*n+sin(x+5)+exp(5*y)+4+5*i
findsym(f)
findsym(f,2)
findsym(f,3)
```

```
f =      5*n + exp(5*y) + sin(x + 5) + m^2 + 4 + 5*I
ans =  m,n,x,y
ans =  x,y
ans =  x,y,n
```

## 函数factor( )：进行符号多项式的因式分解

```
syms x y;
factor(x^4-y^4+x^2-y^2)
for i=1:8
   disp(factor(x^i-1))
end
```

```
ans =
(x + y)*(x - y)*(x^2 + y^2 + 1)
x – 1
(x - 1)*(x + 1)
(x - 1)*(x^2 + x + 1)
(x - 1)*(x + 1)*(x^2 + 1)
(x - 1)*(x^4 + x^3 + x^2 + x + 1)
(x - 1)*(x + 1)*(x^2 + x + 1)*(x^2 - x + 1)
(x - 1)*(x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)
(x - 1)*(x + 1)*(x^2 + 1)*(x^4 + 1)
```

```
y1=factor(1234567890)
y2=factor(sym('12345678901234567890'))
```

```
y1 =
2    3    3    5    3607    3803

y2 =
2*3^2*5*101*3541*3607*3803*27961
```

## 函数expand( )：进行符号表达式的展开

```
syms x y;
f1=(x-y)^2+(x+y)^3      %多项式展开
g1=expand(f1)
f2=sin(x-3*y)          %三角函数展开
g2=expand(f2)
f3=exp(2*x^2+4*y)      %指数函数展开
g3=expand(f3)
```

```
f1 =    (x - y)^2 + (x + y)^3
g1 =    x^3 + 3*x^2*y + x^2 + 3*x*y^2 - 2*x*y + y^3 + y^2
f2 =    sin(x - 3*y)
g2 =    sin(x)*cos(y)^3 - 3*cos(x)*cos(y)^2*sin(y) -
        3*sin(x)*cos(y)*sin(y)^2 + cos(x)*sin(y)^3
f3 =    exp(2*x^2 + 4*y)
g3 =    exp(4*y)*exp(2*x^2)
```

函数collect( )：进行符号表达式中同类项的合并。

```
syms x y;
f=-1/4*y*exp(-2*x)+2/5*x*exp(-2*x)+3*exp(-2*x)
y1=collect(f,exp(-2*x))
y2=collect(x^2*y+y*x+3*y^2*x^3-2*x^2-2*x)
y3=collect(x^2*y+y*x+3*y^2*x^3-2*x^2-2*x,y)
```

```
f =3/exp(2*x) + (2*x)/(5*exp(2*x)) - y/(4*exp(2*x))
y1 =((2*x)/5 - y/4 + 3)/exp(2*x)
y2 =(3*y^2)*x^3 + (y - 2)*x^2 + (y - 2)*x
y3 =(3*x^3)*y^2 + (x^2 + x)*y - 2*x^2 - 2*x
```

**函数horner( )**：将符号多项式转换成嵌套形式，可以是符号多项式或符号多项式矩阵形式，多项式中的常数项保持不变。

```
syms x y;
f1=x^4+6*x^3+4*x^2-4
g1=horner(f1)
f2=[x^2-2*x+4 x^3-3*x^2+5 ; x^2-4*x+6 4*x^2+6*x-8]
g2=horner(f2)
```

```
f1 =    x^4 + 6*x^3 + 4*x^2 – 4
g1 =    x^2*(x*(x + 6) + 4) – 4
f2 =    [ x^2 - 2*x + 4, x^3 - 3*x^2 + 5]
        [ x^2 - 4*x + 6, 4*x^2 + 6*x - 8]
g2 =    [ x*(x - 2) + 4, x^2*(x - 3) + 5]
        [ x*(x - 4) + 6, x*(4*x + 6) - 8]
```

**函数numden( )**：获取符号表达式的分子和分母

[n, d]=numden(A)，其中n为分子，d为分母

```
syms x y;
[n1,d1]=numden(sym(3/4))
f=(x/y+3*y/x);
[n2,d2]=numden(f)
```

```
n1 =   3
d1 =   4
n2 =   x^2 + 3*y^2
d2 =   x*y
```

先变换，再求分子和分母

## 6.4.3　符号表达式的化简

在MATLAB中，采用函数simplify( )和函数simple( )进行符号表达式的化简：

函数simplify( )是一个具有普遍意义的工具，能够对包含和式、方根、分数的乘方、指数函数、对数函数、三角函数等的表达式进行化简。

R=simplify(S, N)：对输入参数为S的符号表达式或符号矩阵，化简后返回R，N为化简步数，默认值为50。

```
syms x y;
f1=(2*x^4-2*x)/(2*x^2+4*x+2);
sf1=simplify(f1)
f2=2*cos(x)^2-3*sin(x)^2-2;
sf2=simplify(f2,100)
sf3=simplify(sin(x)^2+cos(x)^2)
```

```
sf1 =
        (x*(x^3 - 1))/(x + 1)^2
sf2 =
        -5*sin(x)^2
sf3 =
        1
```

函数simple( )也能进行符号表达式的化简，该方法比函数simplify( )要简单，所得到的结果也比较合理。

simple(S)：对符号表达式S进行化简。如果不指定输出参数，将显示所有使表达式变短的形式，并返回其中最短的一个。

R=simple(S)：对S进行化简，返回值R，不显示化简过程。

[R, how]=simple(S)：返回化简后的表达式R，以及描述化简方法的字符串how

```
syms x;
simple(cos(x)^2-sin(x)^2)
```

```
simplify:
cos(2*x)
radsimp:
cos(x)^2 - sin(x)^2
simplify(100):
cos(2*x)
… … … …
ans =
cos(2*x)
```

返回使用不同方法得到的化简形式

```
syms x;
[r1,how1]=simple(cos(3*acos(2*x)))
[r2,how2]=simple(cos(x)^2-sin(x)^2)
[r3,how3]=simple(cos(2*x)^2+sin(2*x)^2)
```

```
r1 =        32*x^3 - 6*x
how1 =      simplify(100)
r2 =        cos(2*x)
how2 =      simplify
r3 =        1
how3 =      simplify
```

## 6.4.4 符号表达式的替换

在MATLAB中，可以采用函数subexpr( )和函数subs( )进行符号替换：

函数subexpr( )将符号表达式中重复出现的字符串用变量代替。

[Y, sigma]=subexpr(S, sig)：该函数指定用于变量sig的值，来代替符号表达式中重复出现的字符串。进行替换后，函数的返回至为Y，被替换的符号变量由sigma返回。

sig也可以用`sig`字符或字符串形式

```
syms a x;
t=solve('a*x^3+a*x^2+3*x=0')
[y1,sigma1]=subexpr(t)
sig=a;
[y2,sigma2]=subexpr(t,sig)
[y3,sigma3]=subexpr(t,'abc')
```

```
t =                      0
       -(a + (a^2 - 12*a)^(1/2))/(2*a)
       -(a - (a^2 - 12*a)^(1/2))/(2*a)
y1 =            0
       -(a + sigma)/(2*a)
       -(a - sigma)/(2*a)
sigma1 =      (a^2 - 12*a)^(1/2)
y2 =           0
       -(a + sig)/(2*a)
       -(a - sig)/(2*a)
sigma2 =      (a^2 - 12*a)^(1/2)
y3 =           0
       -(a + abc)/(2*a)
       -(a - abc)/(2*a)
sigma3 =       (a^2 - 12*a)^(1/2)
```

先对方程求根，
然后进行符号替换

函数subs( )将符号表达式中重复出现的字符串用数值或字符串替换。

Y=subs(S)：将符号表达式S中的所有符号变量，用工作空间中的变量值进行替换；

Y=subs(S, new)：将符号表达式S中的自由符号变量用数值型变量或表达式new进行替换，所有符号变量用工作空间中的变量值进行替换；

Y=subs(S, new, old)：将符号表达式S中的符号变量old用数值型变量或表达式new进行替换

```
syms x y;
f=x^2+3*x+5*x*y+6*y+2*y^2;
g1=subs(f,x,3)
g2=subs(f,3)
g3=subs(f,y,5)
x=5;
g4=subs(f)
```

```
g1 =   2*y^2 + 21*y + 18
g2 =   2*y^2 + 21*y + 18
g3 =   x^2 + 28*x + 80
g4 =   2*y^2 + 31*y + 40
```

```
syms x y;
f=x^2+3*x+5*x*y+6*y+2*y^2
g1=subs(f,{x,y},{3,5})
g2=subs(exp(x*y),x,magic(4))
```

```
f =      x^2 + 5*x*y + 3*x + 2*y^2 + 6*y
g1 =     173
g2 =   [ exp(16*y),  exp(2*y),  exp(3*y), exp(13*y)]
       [  exp(5*y), exp(11*y), exp(10*y),  exp(8*y)]
       [  exp(9*y),  exp(7*y),  exp(6*y), exp(12*y)]
       [  exp(4*y), exp(14*y), exp(15*y),    exp(y)]
```

## 6.4.5 反函数运算

在MATLAB中，采用函数finverse( )进行反函数运算。该函数的调用格式为：

g=finverse(f)：该函数将会计算输入参数f的反函数，其中f为符号表达式，以默认的变量为自变量。函数的返回值g也是一个符号表达式。

g=finverse(f, v)：输入参数v是一个符号变量，是函数f中的变量，且该函数的返回值g以v为自变量。

```
syms x y;
f=sin(x)+cos(y)+5
g1=finverse(f)
g2=finverse(f,x)
g3=finverse(f,y)
```

```
f =      cos(y) + sin(x) + 5
Warning: finverse(cos(y) + sin(x) + 5) is not unique.
g1 =   pi + asin(cos(y) - x + 5)
Warning: finverse(cos(y) + sin(x) + 5) is not unique.
g2 =   pi + asin(cos(y) - x + 5)
Warning: finverse(cos(y) + sin(x) + 5) is not unique.
g3 =   acos(y - sin(x) - 5)
```

# 6.4.6　复合函数运算

在数学计算中，经常会遇到求复合函数的情况。例如，函数z=f(y)，而该函数的自变量y又是一个关于x的函数，即y=g(x)。此时，z=f(g(x))，z是关于x的一个复合函数。

在MATLAB中，采用函数compose( )进行符合函数运算：

compose(f, g)：返回函数当f=f(x) 和 g=g(y) 时的复合函数f(g(y))；

compose(f, g, z)：返回函数当f=f(x) 和 g=g(y) 时的复合函数f(g(z))，返回函数以z为自变量。

**compose(f, g, x, z)：返回复合函数f(g(z))，x为函数f的独立变量；**

**compose(f, g, x, y, z)：返回复合函数f(g(z))，x为函数f的独立变量，y为函数g的独立变量；**

```
syms x y z t u;
f=cos(x/t);
g=sin(y/u);
c1=compose(f,g)
c2=compose(f,g,z)
c3=compose(f,g,x,z)
c4=compose(f,g,t,z)
c5=compose(f,g,x,y,z)
c6=compose(f,g,x,u,z)
c7=compose(f,g,t,u,z)
```

```
c1 =    cos(sin(y/u)/t)
c2 =    cos(sin(z/u)/t)
c3 =    cos(sin(z/u)/t)
c4 =    cos(x/sin(z/u))
c5 =    cos(sin(z/u)/t)
c6 =    cos(sin(y/z)/t)
c7 =    cos(x/sin(y/z))
```

# 6.5　符号矩阵的计算

符号矩阵也是一种符号表达式，前面介绍的符号表达式运算都可以用于符号矩阵，需要注意的是这些函数作用于符号矩阵时，是分别作用于矩阵的每一个元素。由于符号矩阵是一个矩阵，所以符号矩阵还能进行有关矩阵的运算。此外，曾介绍过的许多应用于数值矩阵的函数，也可直接应用于符号矩阵。

# 6.5.1　符号矩阵的生成

在MATLAB中，可以采用函数sym( )直接产生符号矩阵，也可以将数值矩阵转换为符号矩阵。

1. 采用函数sym( )直接生成符号矩阵
2. 利用函数sym( )将数值型变量转换为符号型变量

## 采用函数sym( )直接生成符号矩阵

符号矩阵的元素之间使用空格或逗号进行分隔，各符号表达式的长度可以不同。

```
syms x;
A1=sym('[1.5 x sin(x) ; cos(3)*2/4 4*x exp(x)]')
size(A1)
A2=sym('[1/4, 2/6, 3/5 ; 1/3, 8/9, 8/4 ; 2/3, 3/5, 5/7]')
size(A2)
```

```
A1 =    [       1.5,   x, sin(x)]
        [ cos(3)/2, 4*x, exp(x)]
ans =      2          3
A2 =    [ 1/4, 1/3, 3/5]
        [ 1/3, 8/9,   2]
        [ 2/3, 3/5, 5/7]
ans =      3          3
```

## 利用函数sym( )将数值型变量转换为符号型变量

在matlab中，数值型变量和符号型变量是两种不同类型的数据，不能直接进行计算，系统会将数值型变量转换为符号型变量，然后再进行计算，计算结果为符号型变量。

也可以通过sym函数将数值型变量转换为符号型。

```
syms x;
A1=sym('[1.5 x sin(x) ; cos(3)*2/4 4*x exp(x)]')
a=double(5);
A2=A1+a
M1=magic(4)
A3=sym(M1)
```

```
A1 =  [      1.5,      x,       sin(x)]
      [ cos(3)/2,  4*x,      exp(x)]
A2 =  [          6.5,          x + 5,      sin(x) + 5]
      [ cos(3)/2 + 5,    4*x + 5,   exp(x) + 5]
M1 =  16    2    3   13
       5   11   10    8
       9    7    6   12
       4   14   15    1
A3 =  [ 16,  2,  3, 13]
      [  5, 11, 10,  8]
      [  9,  7,  6, 12]
      [  4, 14, 15,  1]
```

# 6.5.2　符号矩阵的四则运算

对于符号变量和的四则运算有：

**A+B**和**A-B**：实现符号矩阵的加法和减法。如果**A**和**B**为同类型的矩阵（具有相同的行数和列数）时，分别对对应的元素进行加减运算。如果**A**和**B**中有一个为标量，则矩阵中的每一个元素和该标量进行加法或减法。

**A*B**：实现矩阵**A**和**B**的乘法，要求矩阵**A**的列数必须等于矩阵**B**的行数。如果**A**和**B**中有一个为标量，则将矩阵中的每一个元素乘以该标量。

```
syms x;
A1=sym('[1/2 2/5 3 ; 2 3/7 4 ; 1.4 6 7]')
B1=sym(magic(3))
C1=A1+B1
C2=A1-B1
C3=A1+5
D1=A1*B1
D2=A1*5
```

```
A1 =    [ 1/2, 2/5, 3]
        [   2, 3/7, 4]
        [ 1.4,   6, 7]
B1 =    [ 8, 1, 6]
        [ 3, 5, 7]
        [ 4, 9, 2]
C1 =    [ 17/2,  7/5,  9]
        [    5, 38/7, 11]
        [  5.4,   15,  9]
C2 =    [ -15/2,  -3/5, -3]
        [    -1, -32/7, -3]
        [  -2.6,    -3,  5]
C3 =    [ 11/2, 27/5,  8]
        [    7, 38/7,  9]
        [  6.4,   11, 12]
D1 =    [  86/5,  59/2, 59/5]
        [ 233/7, 281/7,   23]
        [  57.2,  94.4, 64.4]
D2 =    [ 5/2,    2, 15]
        [  10, 15/7, 20]
        [ 7.0,   30, 35]
```

**A\B：实现矩阵的左除法。X=A\B为符号线性方程组A*X=B的解。另外，A\B近似等于inv(A)*B。如果X不存在或不唯一，则系统显示警告信息。**

**B/A：实现矩阵的右除法。X=B/A为符号线性方程组X*A=B的解。B/A近似为B*inv(A)。如果X不存在或不唯一，则系统显示警告信息。**

```
syms x;
A=sym('[2 5 3 ; 1 6 4 ; 5 6 7]')
B=sym(magic(3))
C1=A\B                %左除
C2=inv(A)*B
D1=B/A                %右除
D2=B*inv(A)
```

```
A =    [ 2, 5, 3]
       [ 1, 6, 4]
       [ 5, 6, 7]
B =    [ 8, 1, 6]
       [ 3, 5, 7]
       [ 4, 9, 2]
C1 =   [  101/29, -49/29,  -7/29]
       [   81/29, -37/29,  61/29]
       [ -125/29, 104/29, -39/29]
C2 =   [  101/29, -49/29,  -7/29]
       [   81/29, -37/29,  61/29]
       [ -125/29, 104/29, -39/29]
D1 =   [  13/29, -59/29,  53/29]
       [ -49/29,  35/29,  30/29]
       [ 141/29, -51/29, -23/29]
D2 =   [  13/29, -59/29,  53/29]
       [ -49/29,  35/29,  30/29]
       [ 141/29, -51/29, -23/29]
```

## 6.5.3　符号矩阵的线性代数运算

符号矩阵和数值矩阵非常的类似，也可以求符号矩阵的转置、秩、逆矩阵、行列式和特征值分解等。

- 符号矩阵的转置
- 符号方阵的幂运算
- 符号矩阵的秩
- 符号方阵的逆矩阵和行列式计算
- 符号方阵的特征值分解
- 约当标准型
- 符号矩阵的奇异值分解

## ➢ 符号矩阵的转置

符号矩阵A的Hermition转置为A’，也可以用函数ctranspose(A)。如果A为复数矩阵，则A’为复数矩阵的共轭转置；

符号矩阵A的真正转置为A.’，即使为复数，也不进行共轭转置。A.’也可以采用函数transpose(A)来实现。

```
A1=sym(magic(3))
B1=A1'
C1=A1.'
A2=sym([2+3i, 3 ; 2-5i, 8])
B2=A2'
C2=A2.'
```

```
A1 =    [ 8, 1, 6]
        [ 3, 5, 7]
        [ 4, 9, 2]
B1 =    [ 8, 3, 4]
        [ 1, 5, 9]
        [ 6, 7, 2]
C1 =    [ 8, 3, 4]
        [ 1, 5, 9]
        [ 6, 7, 2]
A2 =    [ 3*i + 2, 3]
        [ 2 - 5*i, 8]
B2 =    [ 2 - 3*i, 5*i + 2]
        [       3,       8]
C2 =    [ 3*i + 2, 2 - 5*i]
        [       3,       8]
```

## ➢ 符号矩阵的幂运算

符号矩阵的幂运算为$A^B$，计算矩阵A的整数B次幂。

如果A为标量，而B为方阵，则用$A^B$计算方阵B的特征值和特征向量计算数值。

如果A和B同时为方阵，则返回错误信息。

```
a=sym('[x 2*x 3 ; 3 8 x ; 2.0 x 6]')
y1=a^2
b=sym('[2 4 ; 3 1]')
y2=2^b
```

```
a =   [   x, 2*x, 3]
      [   3,   8, x]
      [ 2.0,   x, 6]
y1 =  [ x^2 + 6*x + 6.0,   2*x^2 + 19*x, 2*x^2 + 3*x + 18]
      [     5.0*x + 24, x^2 + 6*x + 64,          14*x + 9]
      [   5.0*x + 12.0,         18.0*x,      x^2 + 42.0]
b =   [ 2, 4]
      [ 3, 1]
y2 =  [ 515/28, 127/7]
      [ 381/28,  97/7]
```

## ➢ 符号矩阵的秩

利用函数rank( )来计算符号矩阵的秩：

**rank(A)**：求符号矩阵的秩，秩为矩阵A中线性无关的行或列的个数；

**rank(A, tol)**：求比tol值大的值的个数。默认的tol值为max(size(A))*norm(A)*eps

```
syms x y;
f1=sym('[1,x^2,3;exp(x),x+y,y;3+x,sin(x),cos(y)]')
f2=sym('[1,x^2,3;exp(x),x+y,y]')
g1=rank(f1)
g2=rank(f2)
```

```
f1 =  [      1,    x^2,      3]
      [ exp(x),  x + y,      y]
      [  x + 3, sin(x), cos(y)]
f2 =  [      1,   x^2, 3]
      [ exp(x), x + y, y]
g1 =  3
g2 =  2
```

## ➢ 符号矩阵的逆矩阵和行列式运算

采用inv( )求方阵的逆矩阵，当方阵X奇异或范数非常小时，系统将提示出错信息。

采用函数det( )求方阵的行列式。

```
syms x;
A1=sym(magic(3))
Y1=inv(A1)
det(A1)
A2=sym([1-x, x, x-1 ; x, x-2, x+3 ; x, x, 0])
Y2=inv(A2)
det(A2)
```

```
A1 =    [ 8, 1, 6]
        [ 3, 5, 7]
        [ 4, 9, 2]
Y1 =    [  53/360, -13/90,  23/360]
        [ -11/180,   1/45,  19/180]
        [  -7/360,  17/90, -37/360]
ans =   -360
A2 =    [ 1 - x,     x, x - 1]
        [     x, x - 2, x + 3]
        [     x,     x,     0]
Y2 =    [ -(x + 3)/(2*x^2 + 7*x - 5),   (x - 1)/(2*x^2 + 7*x - 5),
        (6*x - 2)/(x*(2*x^2 + 7*x - 5))]
        [  (x + 3)/(2*x^2 + 7*x - 5),  -(x - 1)/(2*x^2 + 7*x - 5),
        (2*x^2 + x - 3)/(x*(2*x^2 + 7*x - 5))]
        [        2/(2*x^2 + 7*x - 5), (2*x - 1)/(2*x^2 + 7*x - 5),
        -(2*x^2 - 3*x + 2)/(x*(2*x^2 + 7*x - 5))]
ans =   2*x^3 + 7*x^2 - 5*x
```

## ➢ 符号方阵的特征值分解

采用函数eig求符号方阵的特征值和特征向量：

E=eig(A)：求符号方阵A的特征值，返回值为由特征值组成的向量

[V, D]=eig(A)：计算符号方阵A的特征值和特征向量，返回值V和D为两个方阵。方阵V的每一列为一个特征向量，方阵D为对角矩阵，对角线上的元素为特征值。

特征值和特征向量应满足 A*V=V*D

```
syms x;
A1=sym(magic(3))
E1=eig(A1)
[V1,D1]=eig(A1)
```

```
A1 =    [ 8, 1, 6]
        [ 3, 5, 7]
        [ 4, 9, 2]
E1 =         15
         -2*6^(1/2)
          2*6^(1/2)
V1 =    [ (2*6^(1/2))/5 - 7/5, - (2*6^(1/2))/5 - 7/5, 1]
        [ 2/5 - (2*6^(1/2))/5,   (2*6^(1/2))/5 + 2/5, 1]
        [                   1,                     1, 1]
D1 =    [ -2*6^(1/2),         0,  0]
        [          0, 2*6^(1/2),  0]
        [          0,         0, 15]
```

## ➢ 约当标准型

对矩阵进行相似变换时，会产生约当标准型，求其标准型就是找一个非奇异矩阵V，使J=V\A*V最接近对角矩阵，矩阵V为转换矩阵。Matlab利用函数jordan( )来计算矩阵的约当标准型：

J=jordan(A)：计算矩阵A的约当标准型J

[J, V]=jordan(A)：除计算约当标准型之外，还返回相应的转换矩阵V。

```
A=sym([1, 2 , 1 ; 1, 2, 1 ; 3, 1, 1])
[V,J]=jordan(A)
V1=vpa(V,8)
J1=vpa(J,8)
J2=vpa(V\A*V,8)
```

```
A =     [ 1, 2, 1]
        [ 1, 2, 1]
        [ 3, 1, 1]
V =     [ 1/4 - 5^(1/2)/4, 5^(1/2)/4 + 1/4, -1/5]
        [ 1/4 - 5^(1/2)/4, 5^(1/2)/4 + 1/4, -2/5]
        [               1,               1,    1]
J =     [ 2 - 5^(1/2),           0, 0]
        [           0, 5^(1/2) + 2, 0]
        [           0,           0, 0]
V1 =    [ -0.30901699, 0.80901699, -0.2]
        [ -0.30901699, 0.80901699, -0.4]
        [         1.0,        1.0,  1.0]
J1 =    [ -0.23606798,        0, 0]
        [           0, 4.236068, 0]
        [           0,        0, 0]
```

## ➢ 符号矩阵的奇异值分解

利用函数svd进行矩阵的奇异值分解：

S=svd(A)：计算矩阵A的奇异值对角矩阵，计算精度由digitis( )决定；

[U, S, V]=svd(A)：输出参数U和V时两个正交矩阵，满足A=U*S*V

```
A=sym(magic(3))
svd(A)
vpa(svd(A))
svd(vpa(A))
[U,S,V]=svd(A)
```

```
A =     [ 8, 1, 6]
        [ 3, 5, 7]
        [ 4, 9, 2]
ans =    15
        4*3^(1/2)
        2*3^(1/2)
ans =                        15.0
         6.9282032302755091741097853660235
         3.4641016151377545870548926830117
ans =                        15.0
         6.9282032302755091741097853660235
         3.4641016151377545870548926830117
```

U = [ 0.57735026918962576450914878050196,
0.70710678118654752440084436210485, 0.40824829046386301636621401245098]
[ 0.57735026918962576450914878050196,
2.0101023277448587293827304258269*10^(-50), -0.81649658092772603273242802490196]
[ 0.57735026918962576450914878050196,
-0.70710678118654752440084436210485, 0.40824829046386301636621401245098]

S = [ 15.0, 0, 0]
[ 0, 6.9282032302755091741097853660235, 0]
[ 0, 0, 3.4641016151377545870548926830117]

V = [ 0.57735026918962576450914878050196,
0.40824829046386301636621401245098, 0.70710678118654752440084436210485]
[ 0.57735026918962576450914878050196,
-0.81649658092772603273242802490196, -8.2323241168269385428977070482381*10^(-51)]
[ 0.57735026918962576450914878050196,
0.40824829046386301636621401245098, -0.70710678118654752440084436210485]

# 6.6　符号微积分

## 6.6.1 符号表达式的微分运算

在MATLAB中，采用函数diff( )进行微分和求导运算：

diff( )：没有指定变量和导数阶数，系统按照findsym( )函数指示的默认变量对符号表达式S求一阶导数；

diff(S, n)：对符号表达式S求n阶导数，n为正整数；

diff(S, ‘v’)或diff(S, sym(‘v’))：计算符号表达式S的一阶导数，以符号变量v为自变量；

diff(S, ‘v’, n)：计算符号表达式S的n阶导数，以符号变量v为自变量。

```
syms x y;
f=5*x^4+y*sin(x)+x*cos(y)+6
g1=diff(f)
g2=diff(f,4)
g3=diff(f,x,4)
g4=diff(f,y,4)
```

```
f =x*cos(y) + y*sin(x) + 5*x^4 + 6
g1 =cos(y) + y*cos(x) + 20*x^3
g2 =y*sin(x) + 120
g3 =y*sin(x) + 120
g4 =x*cos(y)
```

## 6.6.2 符号表达式的极限

**MATLAB中，采用函数limit( )求符号表达式的极限：**

**limit(f)：求函数f的极限值，符号函数f的变量为函数findsym确定的默认变量。没有指定变量的目标值时，系统默认趋近于0；**

**limit(f, a)：计算变量趋近于a时的函数f的极限值；**

**limit(f, x, a)：计算自变量x趋近于a时的函数f的极限值；**

**limit(f, x, a, 'right')：计算函数f的极限值，right表示变量x从右侧趋近于a，也可用left代替。表示从左侧趋近于a**

```
syms x h;
y1=limit((cos(x+h)-cos(x))/h,h,0)
y2=limit(((x+h)^3-x^3)/h,h,0)
```

```
y1 =
        -sin(x)
y2 =
        3*x^2
```

## 6.6.3 符号表达式的积分

在MATLAB中，采用函数int( )来计算符号表达式的不定积分和定积分。该函数的调用格式为：

R=int(S)：该函数计算S的不定积分，输入参数S可以是符号表达式或符号矩阵。没有指定积分变量和积分阶数，系统按函数findsym( )得到的默认变量对S求不定积分。

R=int(S, v)：该函数对S中指定的符号变量v求不定积分。需要注意的是，函数的返回值R只是其中的一个原函数，后面没有带任意常数C。

**R=int(S, a, b)**：该函数计算S在闭区间[a, b]上的定积分。由于没有指定积分变量，系统按函数findsym( )得到的默认变量计算定积分。a和b分别表示定积分的下限和上限，可以是两个具体的数，也可以是一个符号表达式，还可以是无穷大（inf）。

**R=int(S, v, a, b)**：该函数计算S对于变量v在区间[a, b]上的定积分。

```
syms x y;
f1=cos(x)+cos(y)
g1=int(f1)
g2=int(f1,x)
g3=int(f1,y)
```

```
f1 =    cos(x) + cos(y)
g1 =    sin(x) + x*cos(y)
g2 =    sin(x) + x*cos(y)
g3 =    sin(y) + y*cos(x)
```

```
syms x;
f1=1/x^2+sin(x)
g1=int(f1,1,2)
g2=int(f1,x,1,2)
f2=1/x^2;
g3=int(f2,x,1,+inf)
```

```
f1 =    sin(x) + 1/x^2
g1 =    cos(1) - cos(2) + 1/2
g2 =    cos(1) - cos(2) + 1/2
g3 =    1
```

## 6.6.4 级数的求和

在MATLAB中，采用函数symsum( )进行级数符号的求和。该函数的调用格式为：

symsum(S)：该函数没有指定求和的符号变量，参数S表示级数的通项。

symsum(S, v)：该函数对变量v进行级数求和。

symsum(S, a, b)：该函数对默认变量从a到b进行级数求和。

symsum(S, v, a, b)：该函数对于变量v从a到b进行级数求和。

```
syms x k;
y1=symsum(k)
y2=symsum(k,1,5)
y3=symsum(sin(x)+cos(k),x,1,5)
y4=symsum(k,5,4)
y5=symsum(1/k^2)
y6=symsum(1/k^2,1,Inf)
```

```
y1 =   k^2/2 - k/2
y2 =   15
y3 =  sin(1) + sin(2) + sin(3) + sin(4) + sin(5) + 5*cos(k)
y4 =   0
y5 =   -psi(1, k)
y6 =   pi^2/6
```

%伽玛函数（Gamma函数）

## 6.6.5 泰勒级数

采用函数taylor( )求符号表达式的泰勒级数展开式：

taylor(f)：计算函数f在默认变量等于0处作默认为5阶的泰勒展开式。

taylor(f, n)：计算函数f在默认变量为0处做n-1阶的泰勒展开式，n的默认值为6。

taylor(f, a)：计算函数f在默认变量等于a处做默认为5阶的泰勒展开式，a的默认值为0。

taylor(f, n, a)：计算函数f在默认变量等于a处做n-1阶的泰勒展开式。

taylor(f, n, v, a)：计算函数f在变量等于a处做n-1阶的泰勒展开式，变量变为v。

```
syms x y;
f1=taylor(exp(x))
f2=taylor(exp(x),8)
f3=taylor(exp(x),3,6)
f4=taylor(exp(x),y)
```

```
f1 =    x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1
f2 =    x^7/5040 + x^6/720 + x^5/120 + x^4/24 + x^3/6 +
x^2/2 + x + 1
f3 =    exp(6) + exp(6)*(x - 6) + (exp(6)*(x - 6)^2)/2
f4 =    exp(y) + (exp(y)*(x - y)^2)/2 + (exp(y)*(x - y)^3)/6
+ (exp(y)*(x - y)^4)/24 + (exp(y)*(x - y)^5)/120 +
exp(y)*(x - y)
```

# 6.7　符号表达式积分变换

在MATLAB中，提供了丰富的积分变换函数，本节将介绍MATLAB提供的常见积分变换，主要包括傅立叶变换及其反变换、拉普拉斯变换及其反变换和Z变换及其反变换。

数学中经常利用某种运算先把复杂问题变换为比较简单的问题，然后求解，由此再求其逆运算就可得到原问题的解。在初等数学中，曾经利用取对数运算把数的积或商分别变换为较简单的和、差运算，其计算过程就是这种思想的具体体现。

积分变换，就是通过积分运算，把一个函数变成另一个函数的变换，一般是含有参变量α的积分

$$F(\alpha)=\int_D f(P)K(p,\alpha)dP$$

其中函数**K(P,α)**可因积分变换不同而不同，称为积分变换的核；**D**是给定的积分区域，当积分变量**P**是实变量时,它就是积分区间。当选取不同的积分域和变换核时，就得到不同名称的积分变换。

## 6.7.1 Fourier变换及其反变换

**时域中的信号f(t)与它在频域中的Fourier变换F(w)之间存在如下的关系。**

$$F(\omega)=\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt$$

$$f(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega$$

**在MATLAB中，采用函数fourier( )计算傅立叶变换，采用函数ifourier( )计算傅立叶变换的反变换。**

```
syms t v w x;
F1=fourier(1/t)
F2=fourier(exp(-x^2),x,t)
F3=fourier(exp(-t)*sym('heaviside(t)'),v)
F4=fourier(diff(sym('F(x)')),x,w)
```

```
F1 =    pi*(2*heaviside(-w) - 1)*I
F2 =    pi^(1/2)/exp(t^2/4)
F3 =    1/(1 + v*i)
F4 =    w*transform::fourier(F(x), x, -w)*i
```

```
syms u v w x;
f1=ifourier(w*exp(-3*w)*sym('heaviside(w)'))
f2=ifourier(1/(1+w^2),u)
f3=ifourier(v/(1+w^2),v,u)
f4=ifourier(fourier(sym('f(x)'),x,w),w,x)
```

```
f1 =    1/(2*pi*(- 3 + x*i)^2)
f2 =    ((pi*heaviside(u))/exp(u) + pi*heaviside(-u)*exp(u))/(2*pi)
f3 =    -(dirac(u, 1)*i)/(w^2 + 1)
f4 =    f(x)
```

## 6.7.2 Laplace变换及其反变换

**对于函数f(x)进行拉普拉斯（Laplace）变换的公式为：**

$$L(s)=\int_0^{\infty} f(t)e^{-st}dt$$

**对函数L(s)进行拉普拉斯反变换的公式为：**

$$F(t)=\int_{c-i\infty}^{c+i\infty} L(s)e^{st}ds$$

**在MATLAB中，实现拉普拉斯变换的函数为laplace( )，进行拉普拉斯反变换的函数为ilaplace( )。**

```
syms s t w x a;
f1=laplace(x^3)
f2=laplace(sin(w*x),t)
f3=laplace(exp(a*x),s)
f4=laplace(cos(w*x),w,t)
```

```
f1 =    6/s^4
f2 =    w/(t^2 + w^2)
f3 =    -1/(a - s)
f4 =    t/(t^2 + x^2)
```

```
syms s t a w x y;
f1=ilaplace(1/(s-1))
f2=ilaplace(1/(t^2+1),y)
f3=ilaplace(1/(s-a),y)
f4=ilaplace(y/(y^2+w^2),y,x)
f5=ilaplace(sym('laplace(F(x),x,s)'),s,x)
```

```
f1 =    exp(t)
f2 =    sin(y)
f3 =    exp(a*y)
f4 =    cos(w*x)
f5 =    F(x)
```

## 6.7.3 Z变换及其反变换

对于离散序列f(n)的Z变换为

$$F(z)=\sum_{n=0}^{\infty} f(n) z^{-n}$$

对于F(z)的Z反变换为：

$$f(n)=Z^{-1}\{F(z)\}$$

在MATLAB中，利用函数ztrans( )进行Z变换，利用函数iztrans( )进行Z反变换。

```
syms k n w z;
f1=ztrans(2^n)
f2=ztrans(sin(k*n),w)
f3=ztrans(cos(k*n),k,z)
```

```
f1 =    z/(z - 2)
f2 =    (w*sin(k))/(w^2 - 2*cos(k)*w + 1)
f3 =    (z*(z - cos(n)))/(z^2 - 2*cos(n)*z + 1)
```

```
syms x k n w z;
f1=iztrans(z/(z-2))
f2=iztrans(exp(x/z),z,k)
f3=iztrans(sym('ztrans(f(n),n,z)'),z,n)
```

```
f1 =    2^n
f2 =    x^k/factorial(k)
f3 =    f(n)
```

# 6.8　符号方程求解

符号运算不仅能够求解方程，还可以求解方程组。下面介绍如何利用函数**solve( )**求解符号代数方程组，函数**dsolve( )**求解微分方程组。

## 6.8.1 符号代数方程组的求解

在MATLAB中，利用函数solve( )求解一般符号代数方程组。该函数的调用格式为：

solve('eqn1')：该函数求解单个的方程eqn1。

solve('eqn1', 'eqn2', …, 'eqnN')：该函数用来求解由N个方程组成的方程组。

solve('eqn1', 'eqn2', …, 'eqnN', 'var1', 'var2', …, 'varN')：该函数中var1等表示符号变量，以这些变量为自变量，进行方程的求解。

求解方程ysin(x)=5，分别以x和y为自变量

```
syms x y;
e1=sym('y*sin(x)=5')
g1=solve(e1)
g2=solve(e1,'y')
g3=solve('y*sin(x)=5')
g4=solve('y*sin(x)=5','y')
```

```
e1 =   y*sin(x) = 5
g1 =    pi - asin(5/y)
       asin(5/y)
g2 =   5/sin(x)
g3 =    pi - asin(5/y)
       asin(5/y)
g4 =   5/sin(x)
```

求解方程组：$\begin{cases} x^2+4x+y^2+5y-6=0 \\ x^2+2x+y^2-4=0 \end{cases}$

```
syms x y;
e1=sym('x^2+4*x+y^2+5*y-6=0')
e2=sym('x^2+2*x+y^2-4=0')
g=solve(e1,e2)
g.x
g.y
[x,y]=solve('x^2+4*x+y^2+5*y-6=0','x^2+2*x+y^2-4=0')
```

```
e1 =     x^2 + 4*x + y^2 + 5*y - 6 = 0
e2 =     x^2 + 2*x + y^2 - 4 = 0
g =      x: [2x1 sym]
         y: [2x1 sym]
ans =     (5*129^(1/2))/29 - 21/29
         - (5*129^(1/2))/29 - 21/29
ans =    20/29 - (2*129^(1/2))/29
         (2*129^(1/2))/29 + 20/29
x =      (5*129^(1/2))/29 - 21/29
         - (5*129^(1/2))/29 - 21/29
y =      20/29 - (2*129^(1/2))/29
         (2*129^(1/2))/29 + 20/29
```

显示变量类型

求解方程组：
$$\begin{cases} ax+by-6=0 \\ bx+ay-4=0 \end{cases}$$

```
syms x y a b;
e1=sym('a*x+b*y-6=0')
e2=sym('b*x+a*y-4=0')
g1=solve(e1,e2)
g1.x
g1.y
g2=solve(e1,e2,'a','b')
g2.a
g2.b
```

```
e1 =    a*x + b*y - 6 = 0
e2 =    a*y + b*x - 4 = 0
g1 =        x: [1x1 sym]
            y: [1x1 sym]
ans = (2*(3*a - 2*b))/(a^2 - b^2)
ans = (2*(2*a - 3*b))/(a^2 - b^2)
g2 =        a: [1x1 sym]
            b: [1x1 sym]
ans =  (2*(3*x - 2*y))/(x^2 - y^2)
ans =  (2*(2*x - 3*y))/(x^2 - y^2)
```

求解方程组：
$$\begin{cases} ax + y - 6 = 0 \\ 5x + ay - 4 = 0 \\ 4x + 6y - 8 = 0 \end{cases}$$

```
syms x y a b;
e1=sym('a*x+y-6')
e2=sym('5*x+a*y-4=0')
e3=sym('4*x+6*y-8=0')
[a,x,y]=solve(e1,e2,e3)
```

```
a =      3 + (42^(1/2)*i)/2
         3 - (42^(1/2)*i)/2
x =      28/41 - (6*42^(1/2)*i)/41
         28/41 + (6*42^(1/2)*i)/41
y =      36/41 + (4*42^(1/2)*i)/41
         36/41 - (4*42^(1/2)*i)/41
```

求解方程组：
$$\begin{cases} 5\cos(x+y)+x^2+y^2=6 \\ 5x+y-4=0 \end{cases}$$

```
syms x y;
e1=sym('5*cos(x+y)+x^2+y^2=6')
e2=sym('5*x+y-4=0')
[x,y]=solve(e1,e2)
```

```
e1 =    5*cos(x + y) + x^2 + y^2 = 6
e2 =    5*x + y - 4 = 0
x =     0.14340192743521602472166854681415
y =     3.2829903628239198763916572659292
```

由于系统无法给出符号解，自动给出该方程组的数值解

## 6.8.2 微分方程的求解

在MATLAB中，采用函数dsolve( )进行微分方程的求解。该函数的调用格式为：

dsolve('eq1')：该函数对单个的微分方程eq1求解。

dsolve('eq1, eq2, …', 'cond1, cond2, …', 'v')：该函数对由eq1, eq2等组成的微分方程组求解，初始条件为cond1, cond2等，自变量为v。如果不指定参数v，则系统默认以t为自变量。

在matlab编程时，dy/dt或dy/dx表示y的一阶导数，用Dy表示；二阶导数$d^2y/dt^2$或$d^2y/dx^2$用D2y表示，n阶导数用Dny表示。

求解微分方程：$$\frac{dx}{dt} = -ax$$

```
dsolve('Dx=-a*x')                  %求微分方程
dsolve('Dx=-a*x','x(0)=1')         %初始值
dsolve('Dx=-a*x','x(0)=1','s')     %变量变为s
```

```
ans =  C2/exp(a*t)
ans =  1/exp(a*t)
ans =  1/exp(a*s)
```

求解二阶微分方程： $$\frac{d^2y}{dt^2} = -a^2y$$

```
g1=dsolve('D2y=-a^2*y')
g2=dsolve('D2y=-a^2*y','y(0)=1')
g3=dsolve('D2y=-a^2*y','y(0)=1','y(pi)=1')
```

```
g1 =   C2*exp(a*t*i) + C3*(1/exp(a*t*i))
g2 =   C7*(1/exp(a*t*i)) - exp(a*t*i)*(C7 - 1)
g3 =   - ((1/exp(a*t*i))*(exp(pi*a*i) - 1))/(1/exp(pi*a*i) -
exp(a*i*pi)) + (exp(a*t*i)*(1/exp(pi*a*i) -
1))/(1/exp(pi*a*i) - exp(a*i*pi))
```

求解二阶微分方程：$$\frac{d^2y}{dx^2}+2\frac{dy}{dx}+2y=0$$

```
g1=dsolve('D2y+2*Dy+2*y=0')
g2=dsolve('D2y+2*Dy+2*y=0','y(0)=1')
g3=dsolve('D2y+2*Dy+2*y=0','y(0)=1','Dy(0)=0','x')
```

```
g1 =   (C2*cos(t))/exp(t) + (C3*sin(t))/exp(t)
g2 =   cos(t)/exp(t) + (C7*sin(t))/exp(t)
g3 =   cos(x)/exp(x) + sin(x)/exp(x)
```

求解微分方程组：

$$\begin{cases} \dfrac{dx}{dt} = y \\ \dfrac{dy}{dt} = -x \end{cases}$$

```
[x1,y1]=dsolve('Dx=y','Dy=-x','t')
[x2,y2]=dsolve('Dx=y','Dy=-x','x(0)=3','y(0)=4','t')
```

```
x1 =   C2*cos(t) + C1*sin(t)
y1 =   C1*cos(t) - C2*sin(t)
x2 =   3*cos(t) + 4*sin(t)
y2 =   4*cos(t) - 3*sin(t)
```

# 6.9　符号函数的图形绘制

在MATLAB中，对于符号方程，可以非常方便的绘制其图形，常用的绘图函数包括

| 函数名 | 函数功能 |
|---|---|
| fplot | 一元函数绘图 |
| ezplot | 二元函数绘图 |
| ezpolar | 极坐标系下绘图 |
| ezplot3 | 三维图形 |
| ezmesh | 三维网格图 |
| ezmeshc | 带等值线的三维网格图 |
| ezcontour | 等值线图 |
| ezcontourf | 带填充的等值线图 |
| ezsurf | 三维彩色曲面图 |
| ezsurfc | 带等值线的三维彩色曲面图 |

## 6.9.1 符号函数曲线的绘制

对于一元函数，可以采用函数fplot( )绘制该函数的图形。

对于多元函数，可以采用函数ezplot( )绘制。

此外，还可以采用函数ezpolar( )在极坐标下绘图。

## ➢ 函数fplot

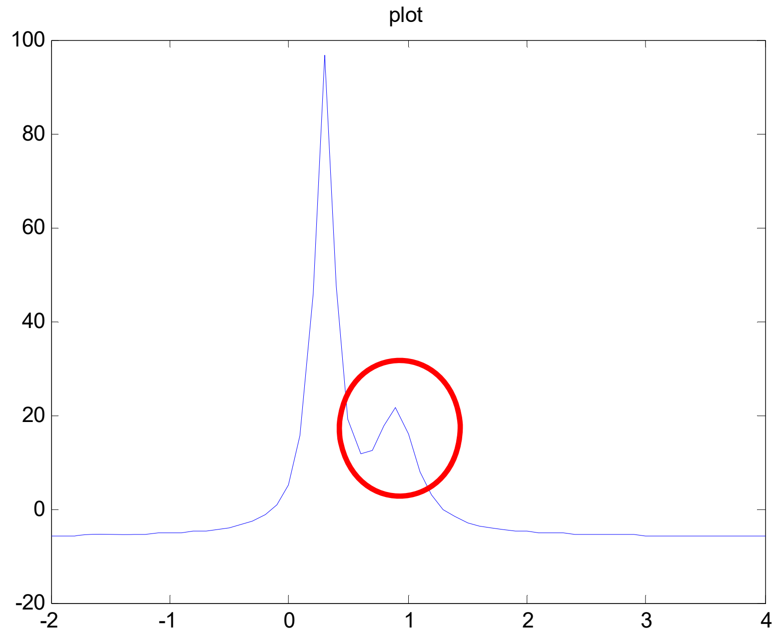
fplot(fun, lims)：直接绘制y=fun(x)的图形，参数lims表示自变量的范围，如果lims只包含2个元素，则表示自变量x的范围，如果lims包含4个元素，则表示x和y的范围；

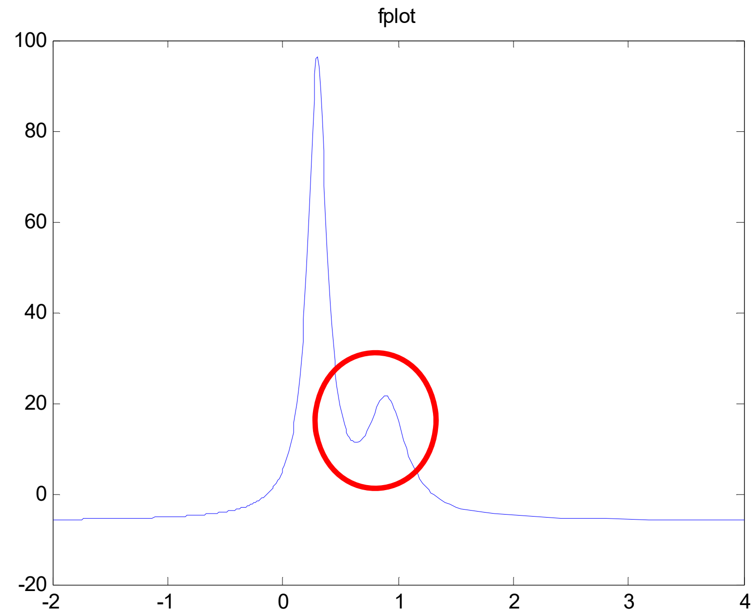
fplot(fun, lims, tol)：该函数中tol<1，代表相对误差，默认值为0.002；

fplot(fun, lims, n)：该函数中n>=1，用最少n+1个点来绘制函数图形，默认值为1；

fplot(fun, lims, ‘LineSpec’)：该函数利用参数LineSpec设置所绘制图形的线性和颜色等。

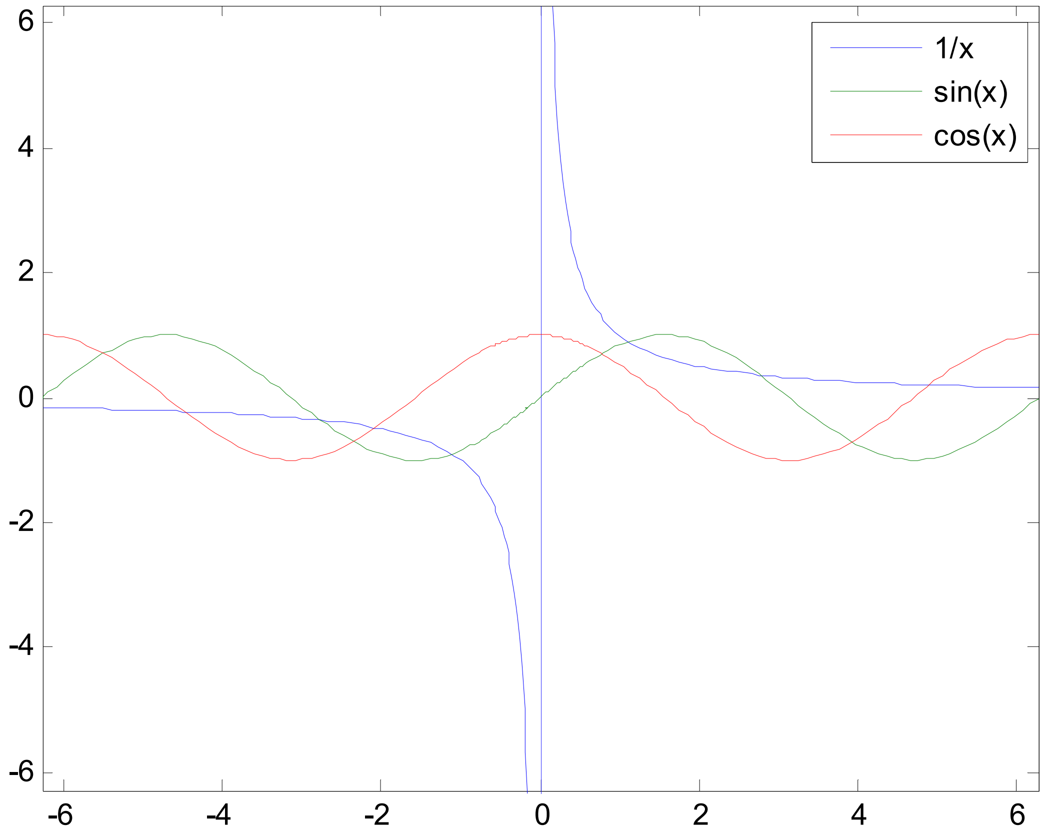
```
x=-2:0.1:4;
figure;
plot(x,humps(x));
title('plot');
figure;
fplot(@humps,[-2 4])
title('fplot');
```



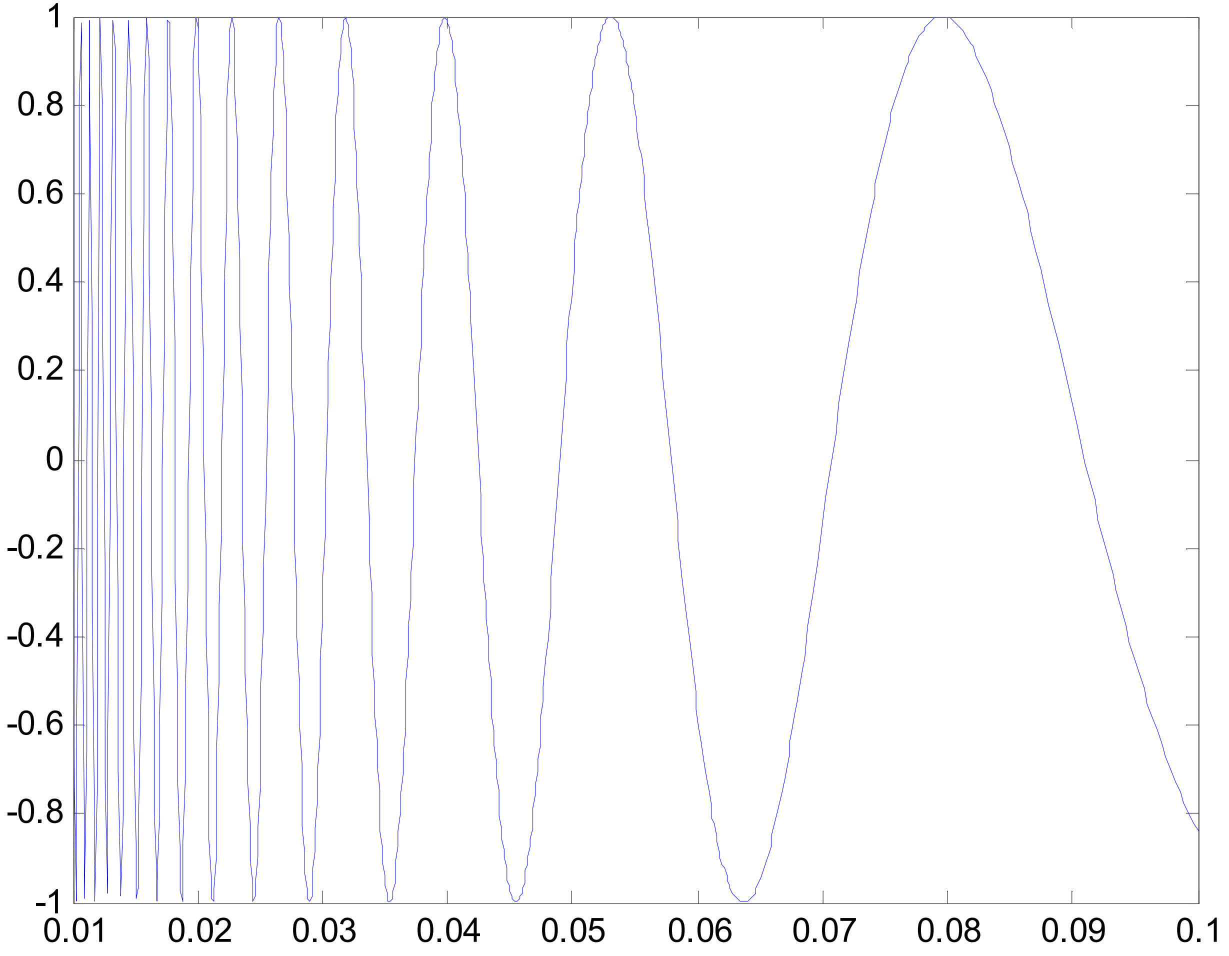




使用**fplot**函数得到的结果更为光滑

```
figure;
fplot('[1/x,sin(x),cos(x)]',2*pi*[-1 1 -1 1]);
legend('1/x','sin(x)','cos(x)');
```



```
figure;
fplot('cos(1/x)',[0.01,0.1],1e-3);
```



➢ 函数ezplot：绘制二元函数

ezplot(fun)：绘制函数y=fun(x)的图形，变量x取值默认为开区间(-2π, 2 π)；

ezplot(fun, [a, b])：变量x取值为开区间(a, b)；

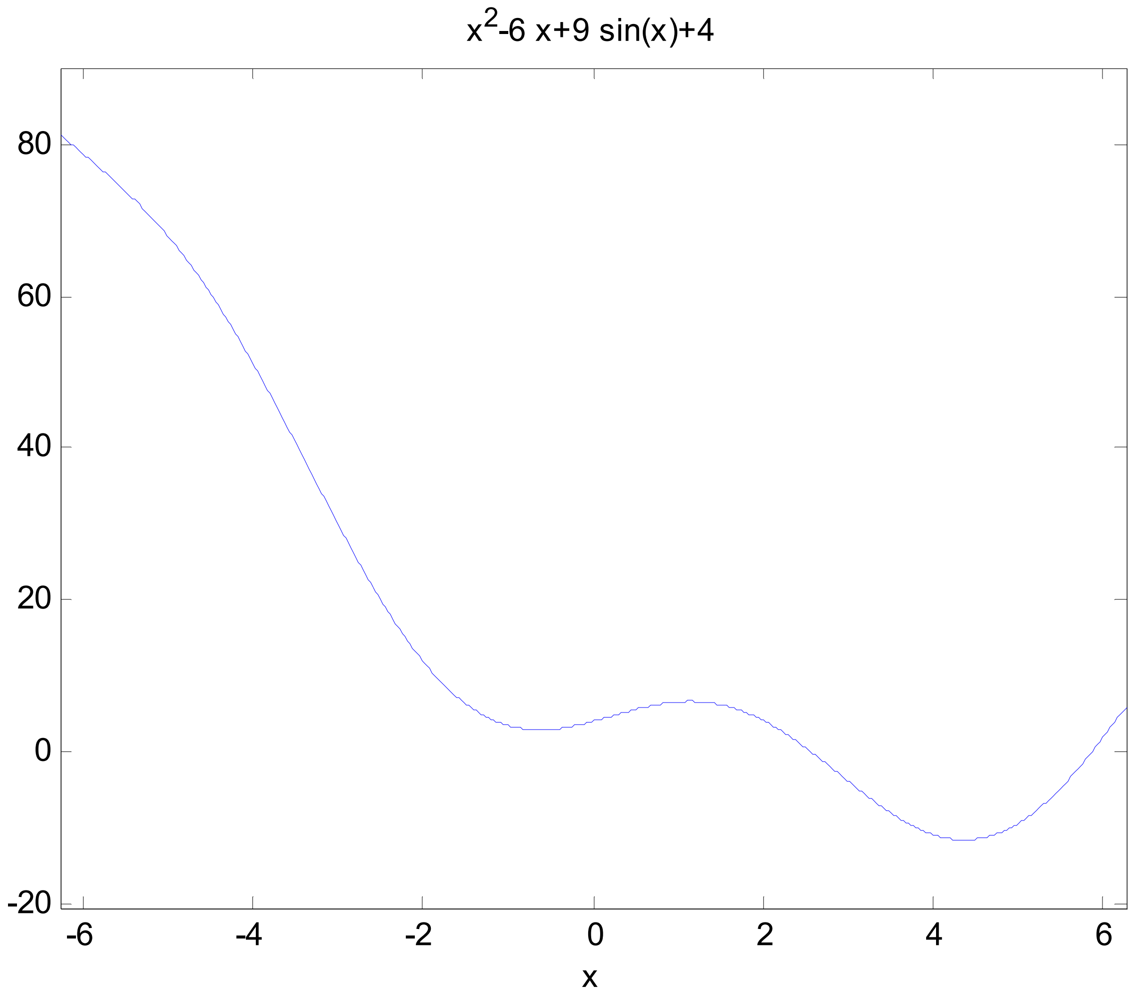
ezpot(fun, [xmin, xmax, ymin, ymax])：设定变量x和y的取值范围；

ezplot(x, y, [tmin, tmax])：在变量t ϵ (tmin, tmax)上绘制函数x=x(t), y=y(t)，t默认取值范围为(-2π, 2 π)；

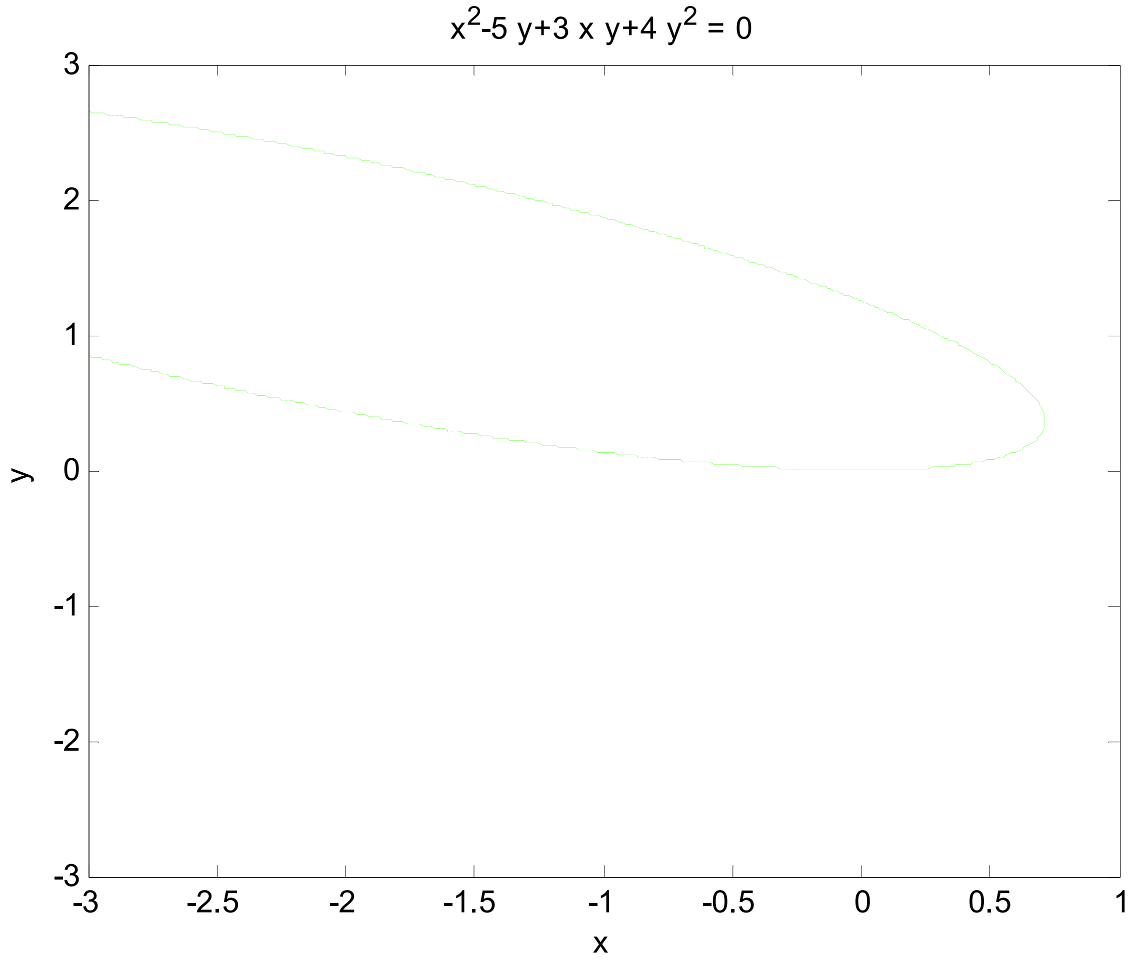
ezplot(x, y, [tmin, tmax], fig)：在指定图形窗口fig中绘制函数图形，默认为当前图形 窗口。

**figure;**

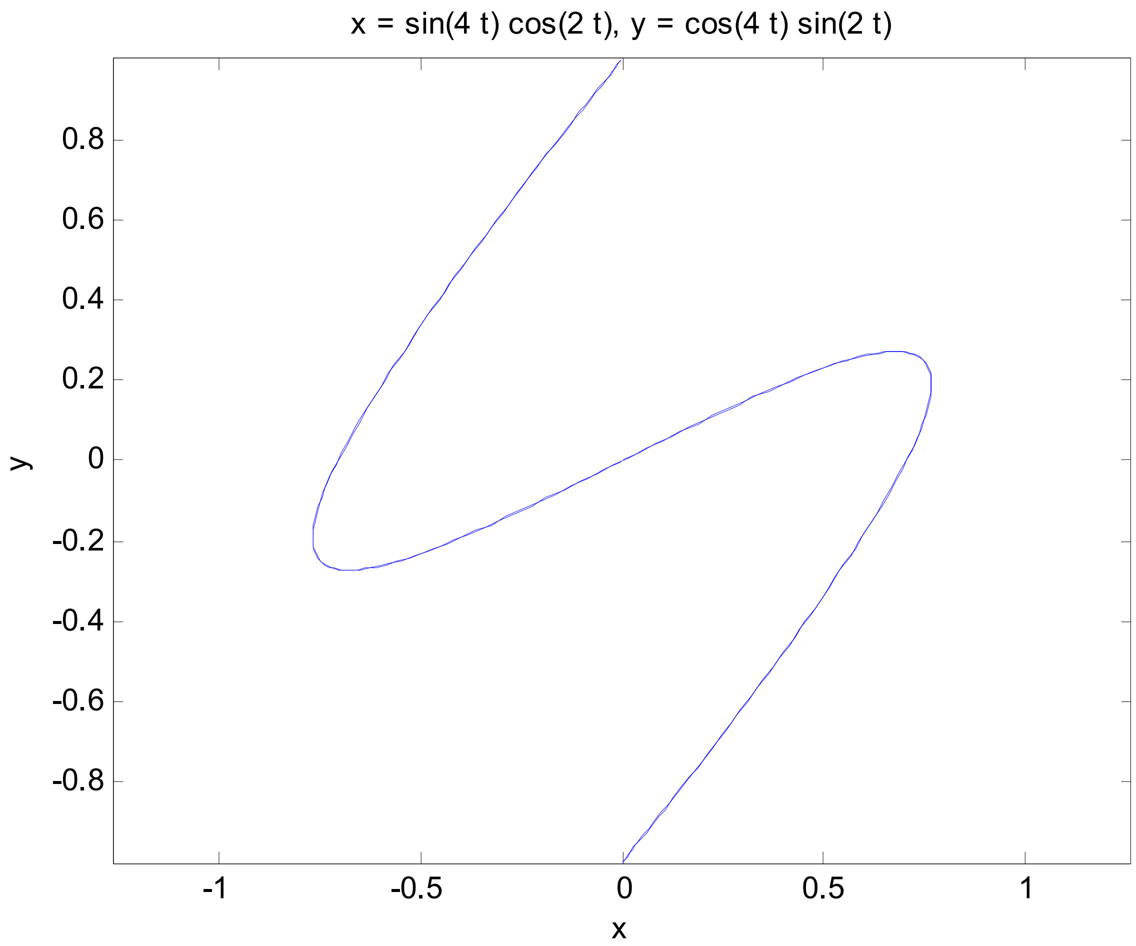
**ezplot('x^2-6*x+9*sin(x)+4',[-2*pi,2*pi]);**



```
figure;
ezplot('x^2-5*y+3*x*y+4*y^2',[-3,1,-3,3]);
```



```
figure;
ezplot('sin(4*t)*cos(2*t)','cos(4*t)*sin(2*t)',[0,pi]);
```

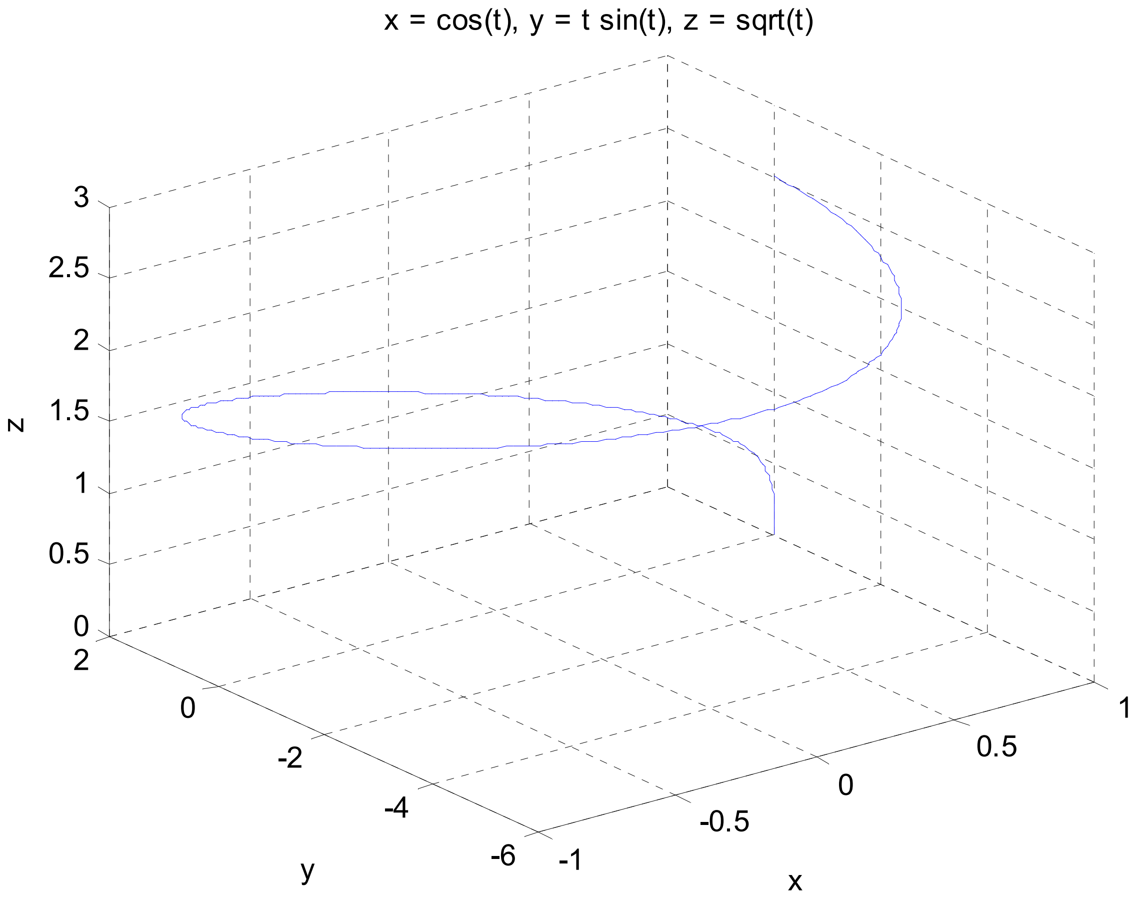


## ➢ 函数ezplot3：绘制符号函数的三维图形

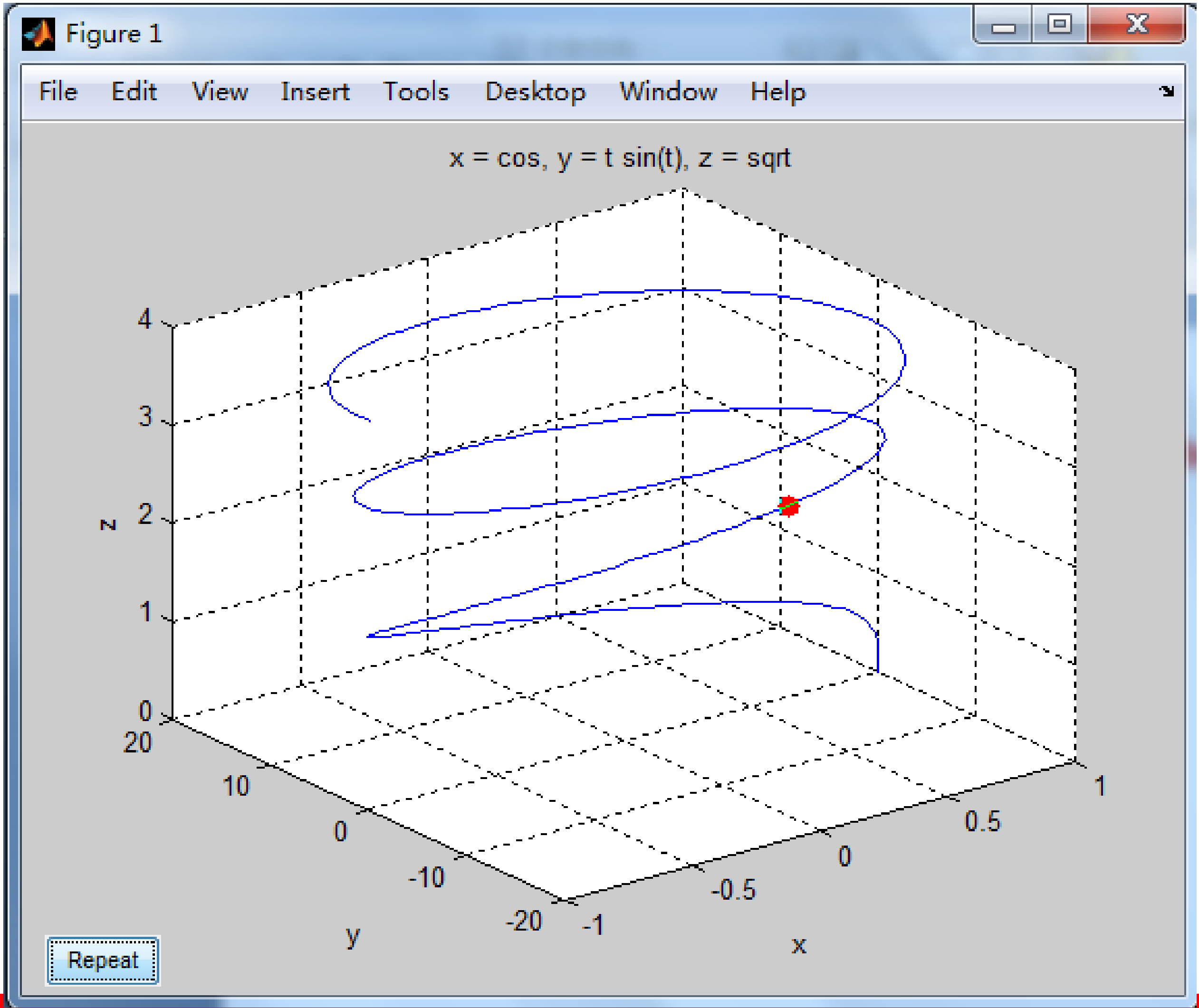
**ezplot(x, y, z, [tmin, tmax])：绘制由函数x=x(t), y=y(t), z=z(t)描述的三维图形，参数t的区间为(tmin, tmax)，默认区间为(-2π, 2 π)；**

**ezplot(x, y, z, [tmin, tmax], ‘animate’)：产生空间曲线的动画效果。**

```
figure;
ezplot3('cos(t)','t*sin(t)','sqrt(t)',[0,2*pi]);
```



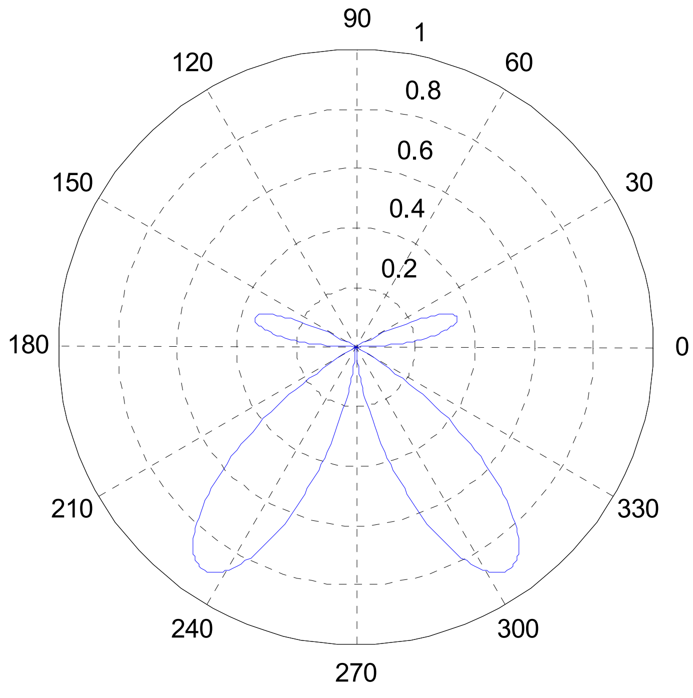
```
figure;
ezplot3(@cos,'t*sin(t)',@sqrt,[0,5*pi],'animate');
```



## ➢ 函数ezpolar：符号函数极坐标系绘图

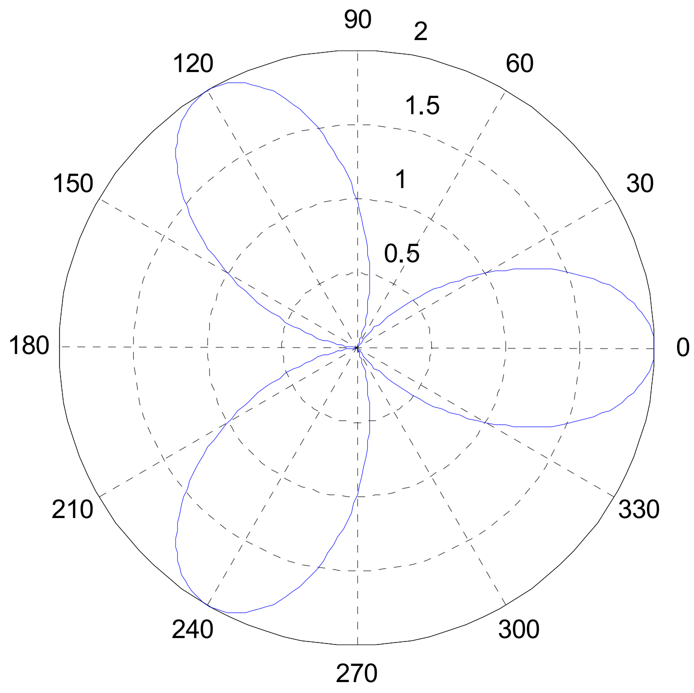
**ezpolar(fun, a, b)：绘制函数rho=fun(theta)，其中theta的取值区间为(a, b)，默认区间为(0, 2π)**

```
figure;
ezpolar('sin(2*t).*cos(3*t)',[0,pi]);
```



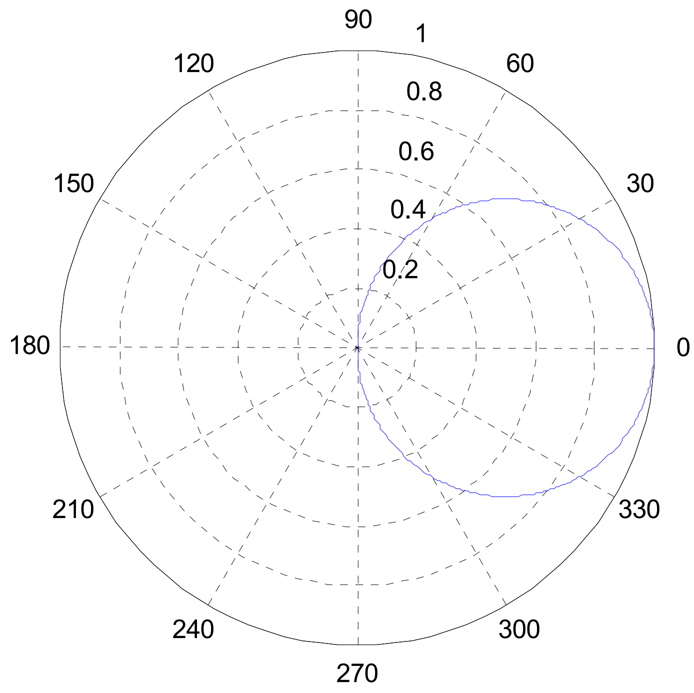
r = sin(2 t) cos(3 t)

```
figure;
ezpolar('1+cos(3*t)',[0,2*pi]);
```




r = 1+cos(3 t)

```
figure;
ezpolar(@cos,[0,pi]);
```




r = cos

## 6.9.2 符号函数的三维网格图

在MATLAB中，可以采用函数ezmesh( )绘制符号函数的三维网格图，利用函数ezmeshc( )绘制带有等值线的三维网格图。

1．函数ezmesh( )

2．函数ezmeshc( )

## ➢ 函数ezmesh：绘制符号函数的三维网格图

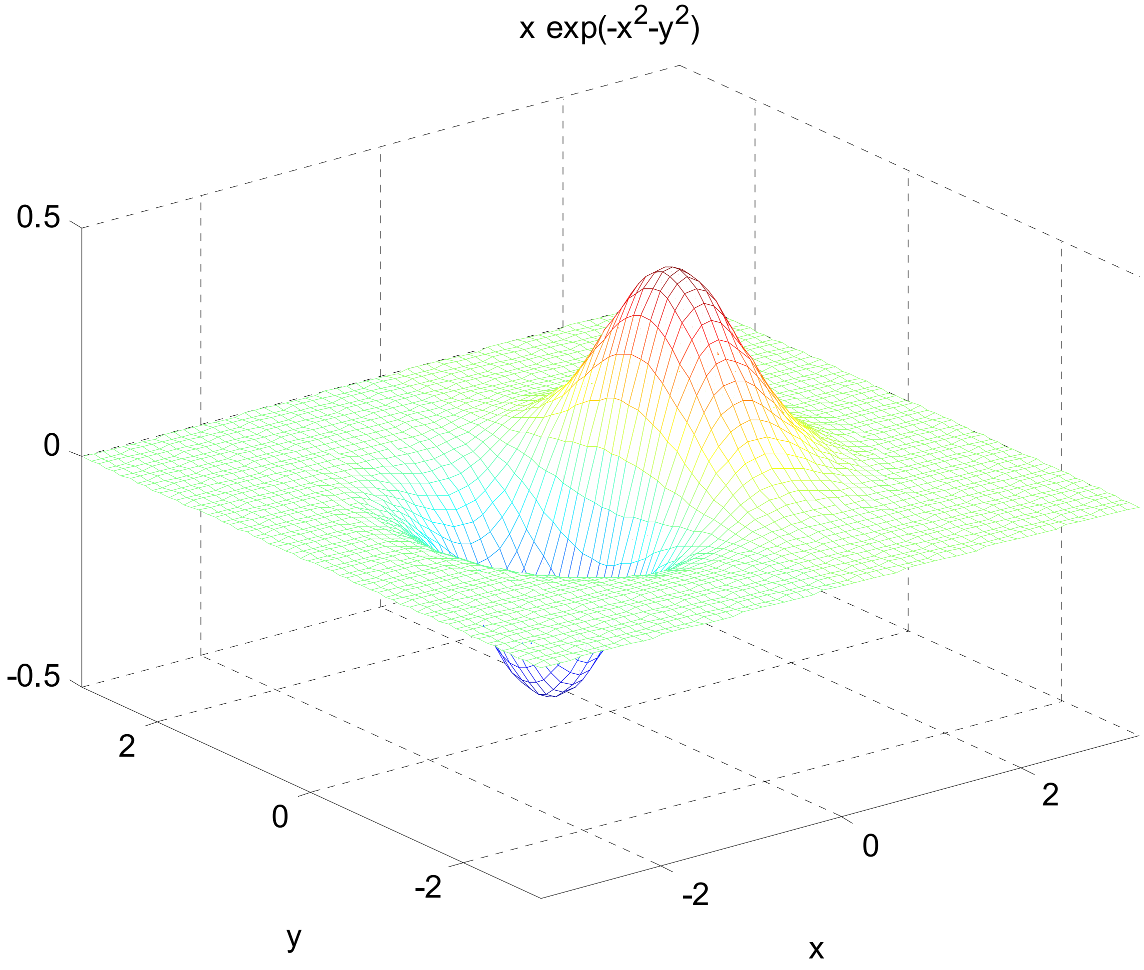
**ezmesh(f, domain)：在domain区间上绘制函数f(x, y)的三维网格图，参数domain包含4个元素[xmin, xmax, ymin, ymax]，也可以包含2个元素，即[a, b]，此时$a<x<b$, $a<y<b$。Domain的默认值为[-2π, 2 π, -2π, 2 π]；**

**ezmesh(x, y, z, [smin, smax, tmin, tmax])：该函数绘制由x=x(s, t), y=y(s, t), z=z(s, t)描述的三维网格图，参数s、t的区间分别为(smin, smax)、(tmin, tmax)；**

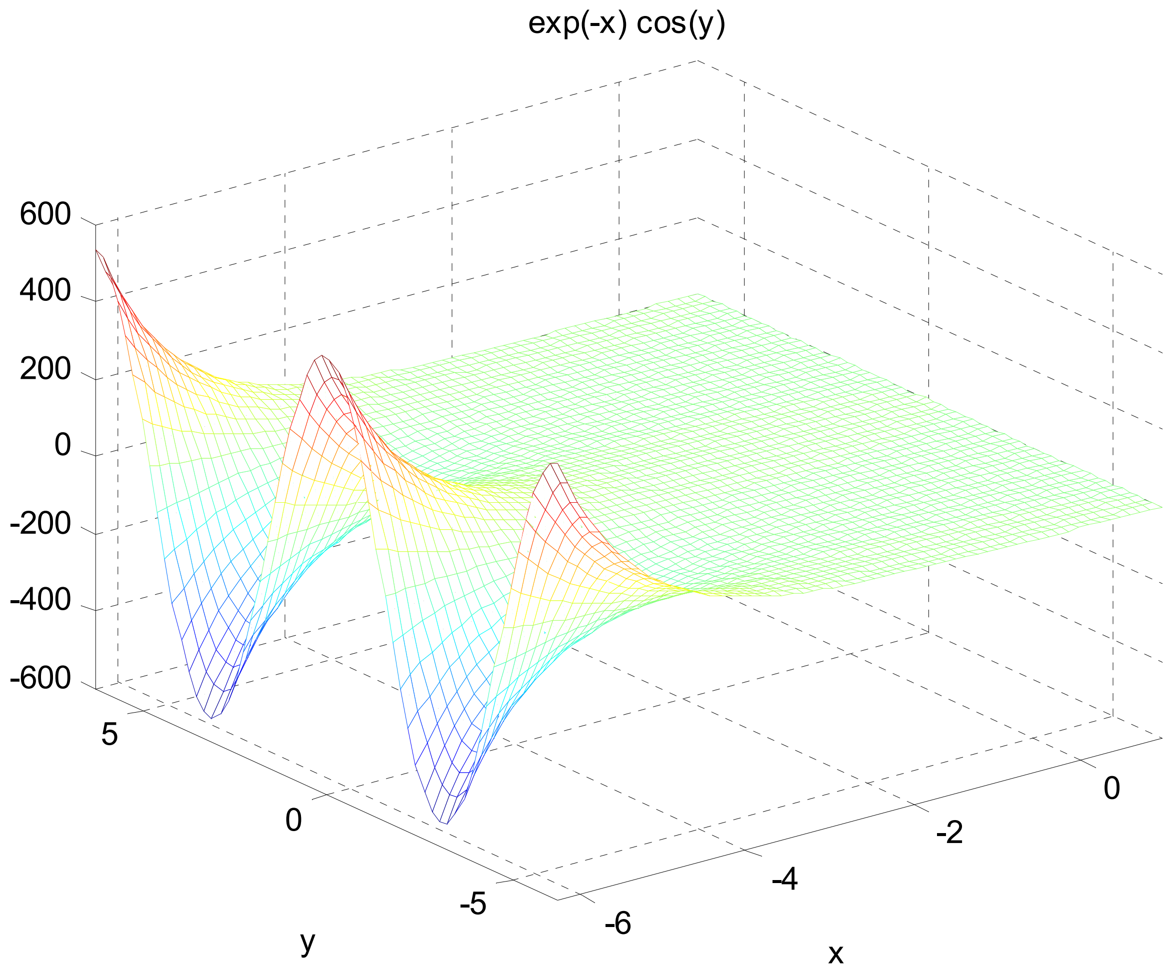
**ezmesh(…, N)：在默认区间上绘制函数f的NxN个网格的网格图，N的默认值为60；**

**ezmesh(…, ‘circ’)：在指定区域的盘面上绘制函数的网格图**

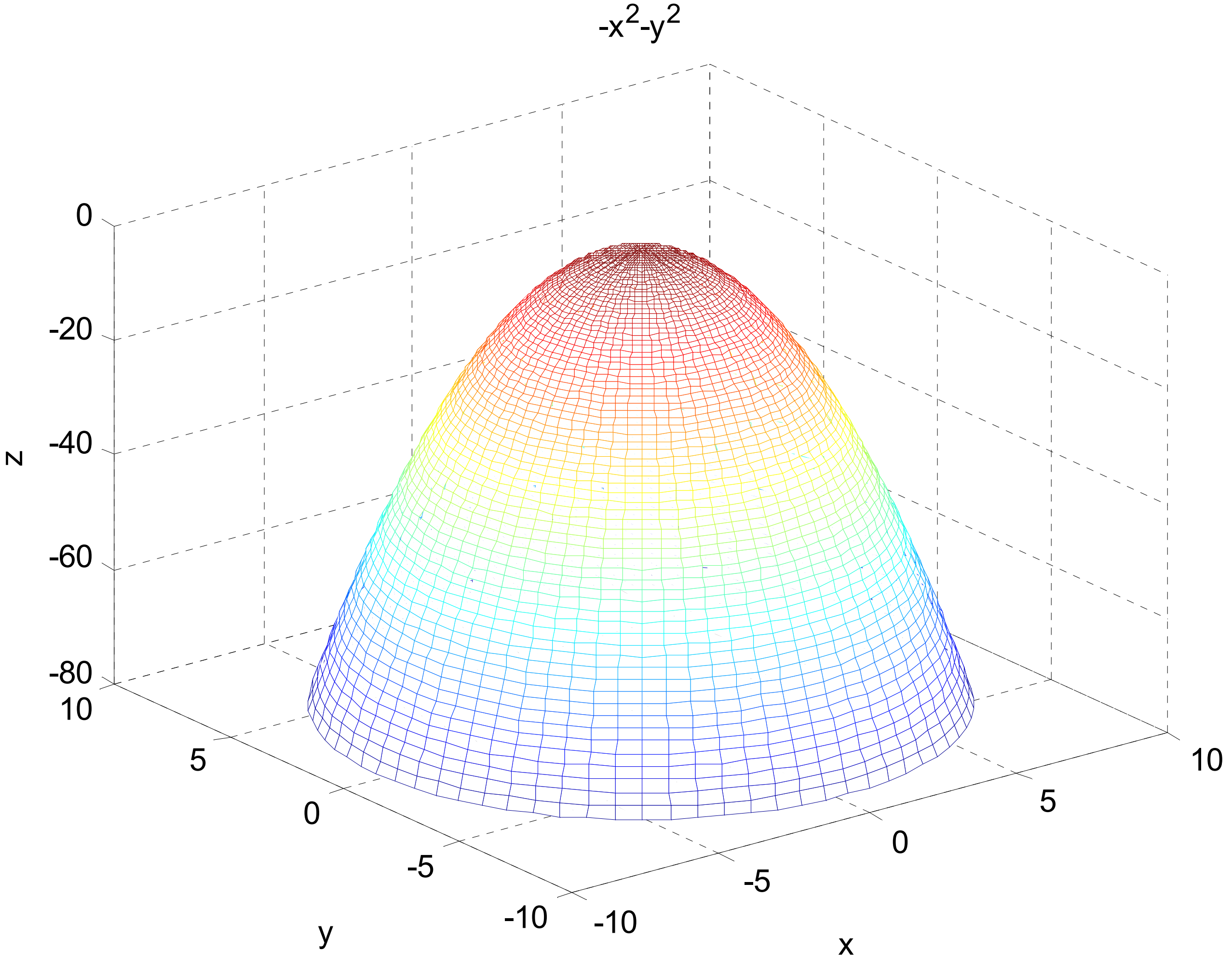
```
figure;
ezmesh('x.*exp(-x.^2-y.^2)');
```



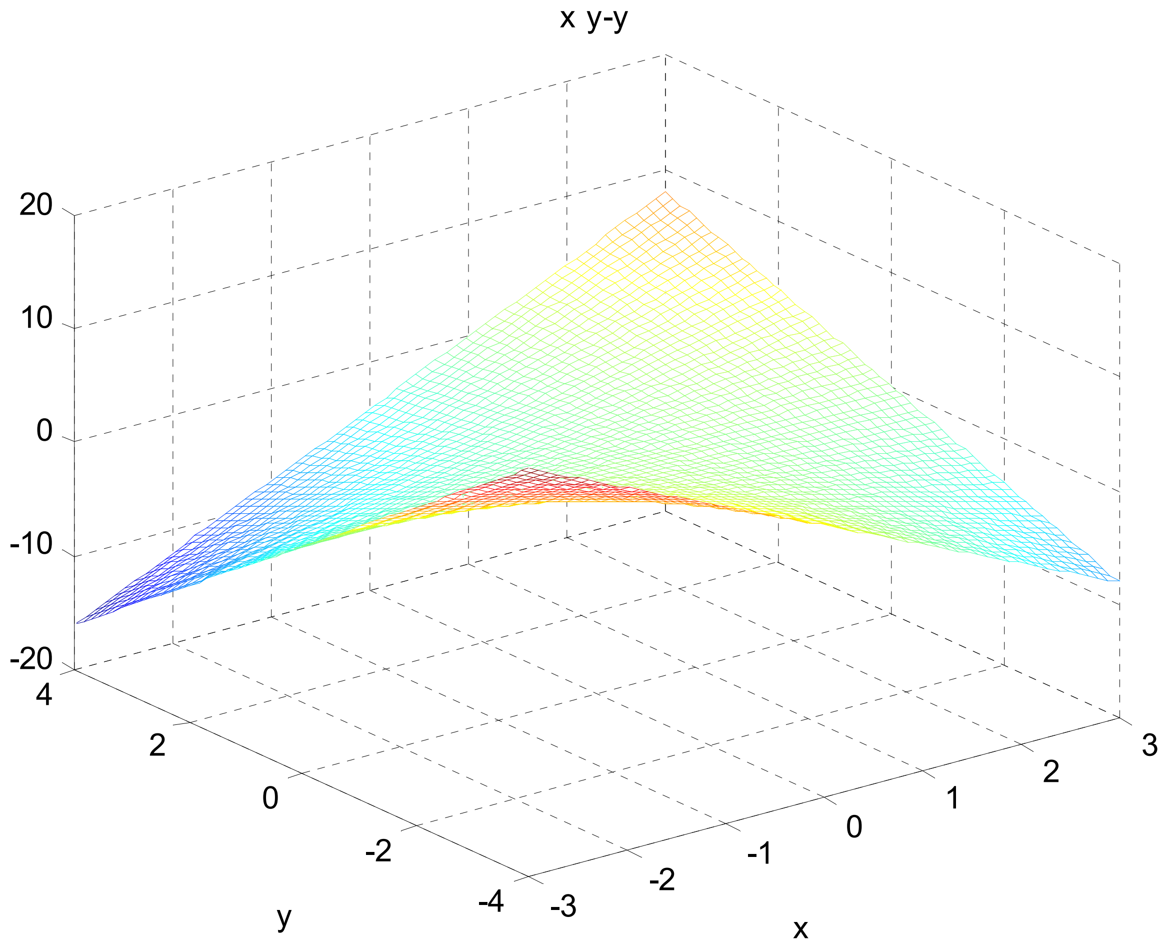
```
figure;
ezmesh('exp(-x)*cos(y)');
```



```
figure;
ezmesh('-x.^2-y.^2','circ');
```

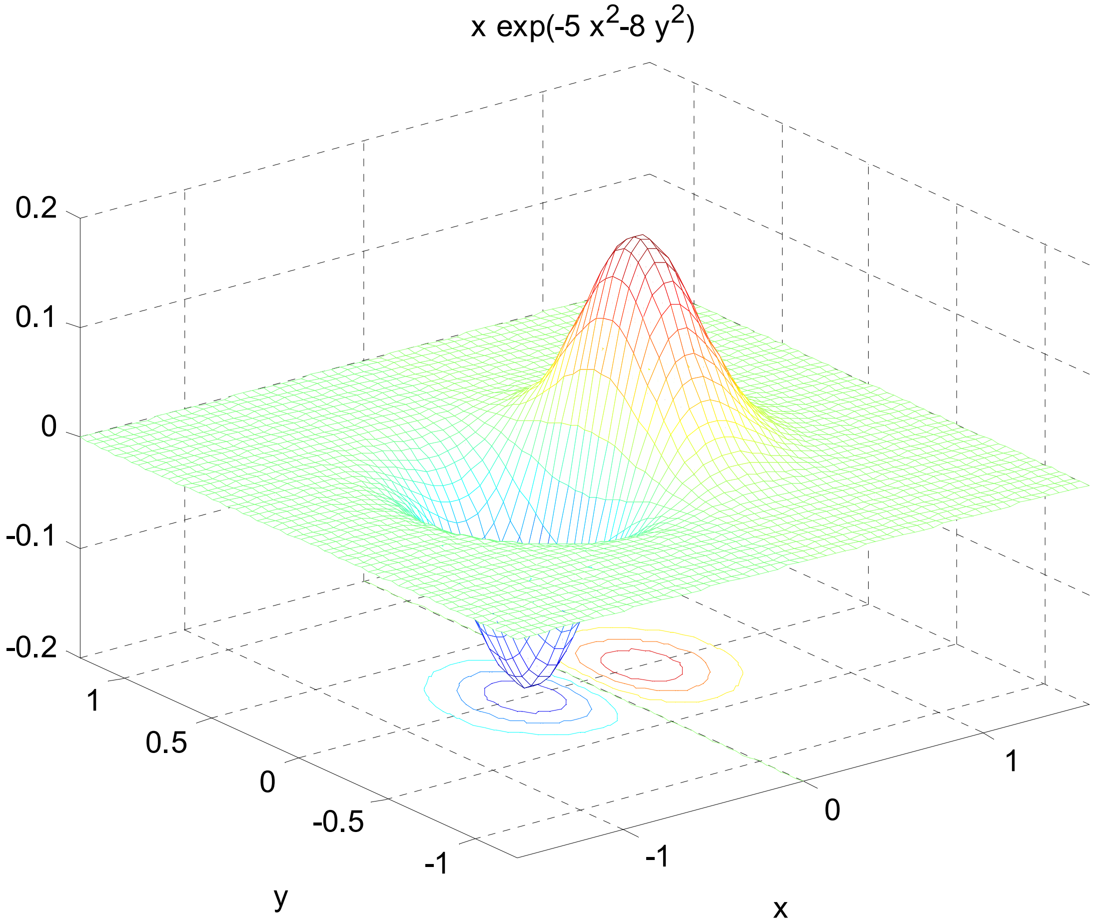


```
figure;
ezmesh('x*y-y',[-3,3,-4,4]);
```

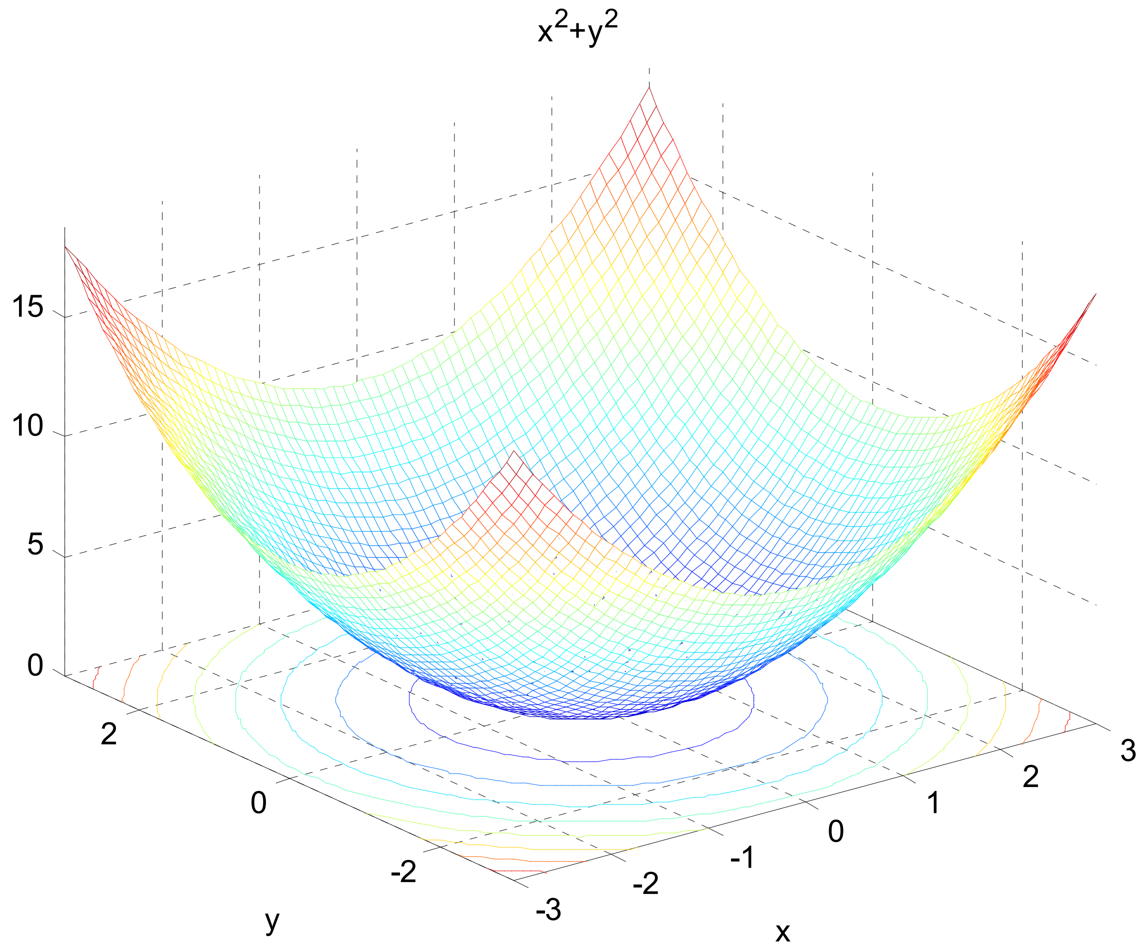


## ➢ 函数ezmeshc：绘制带有等值线的符号函数的三维网格图

```
figure;
ezmeshc('x.*exp(-5*x.^2-8*y.^2)');
```



```
figure;
ezmeshc('x.^2+y.^2',[-3 3 -3 3]);
```



## 6.9.3 符号函数的等值线图

在MATLAB中，可以采用函数ezcontour( )绘制等值线图，利用函数ezcontourf( )绘制经过填充的等值线图。

1．函数ezcontour( )

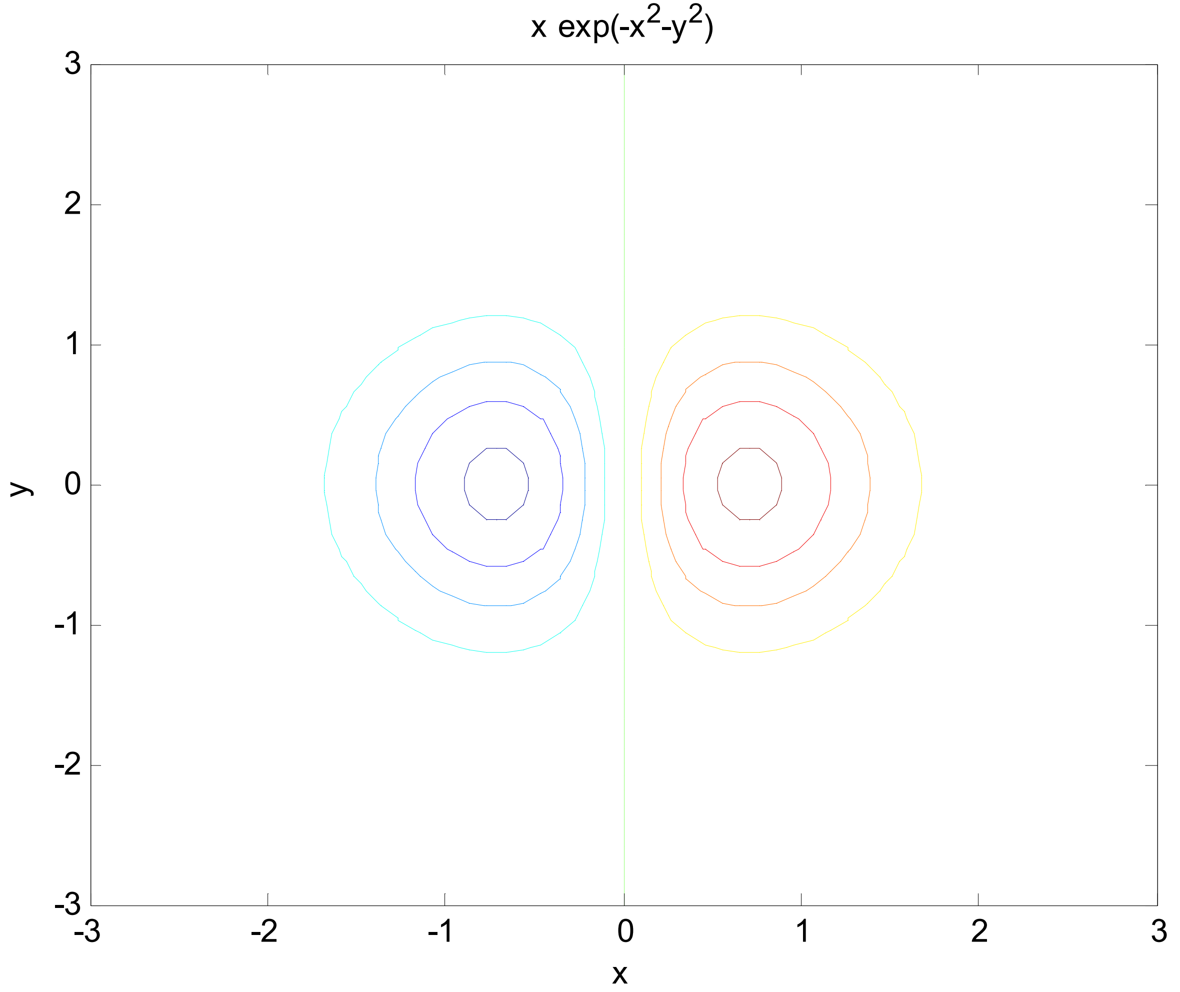
2．函数ezcontourf( )

## ➢ 函数ezcontour：绘制符号函数的等值线图

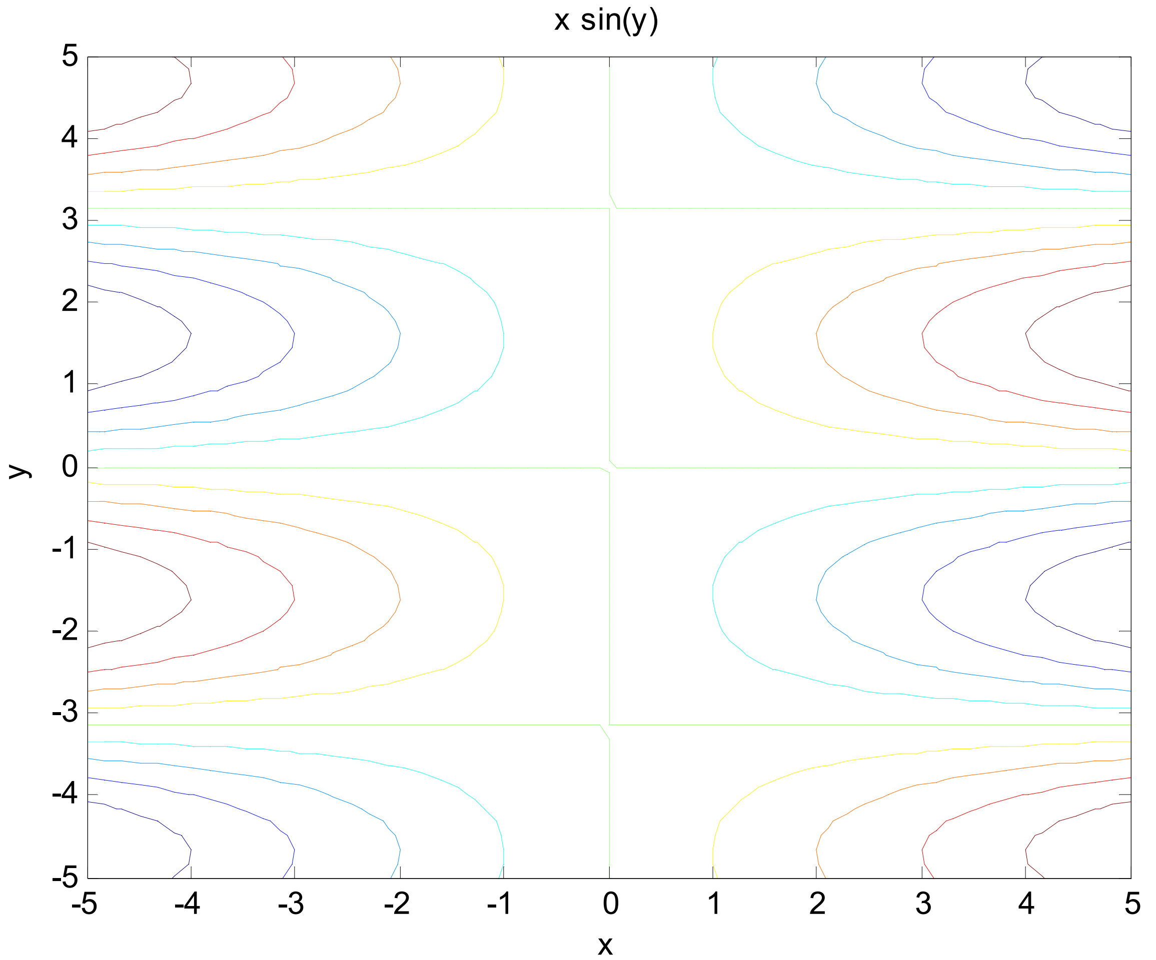
**ezmesh(f, domain)：在domain区间上绘制函数f(x, y)的等值线图，参数domain包含4个元素[xmin, xmax, ymin, ymax]，也可以包含2个元素，即[a, b]，此时$a<x<b$, $a<y<b$。Domain的默认值为[-2π, 2 π, -2π, 2 π]；**

**ezmesh(…, N)：在默认区间上绘制函数f的NxN个网格的等值线图，N的默认值为60；**

```
figure;
ezcontour('x.*exp(-x.^2-y.^2)',[-3 3]);
```

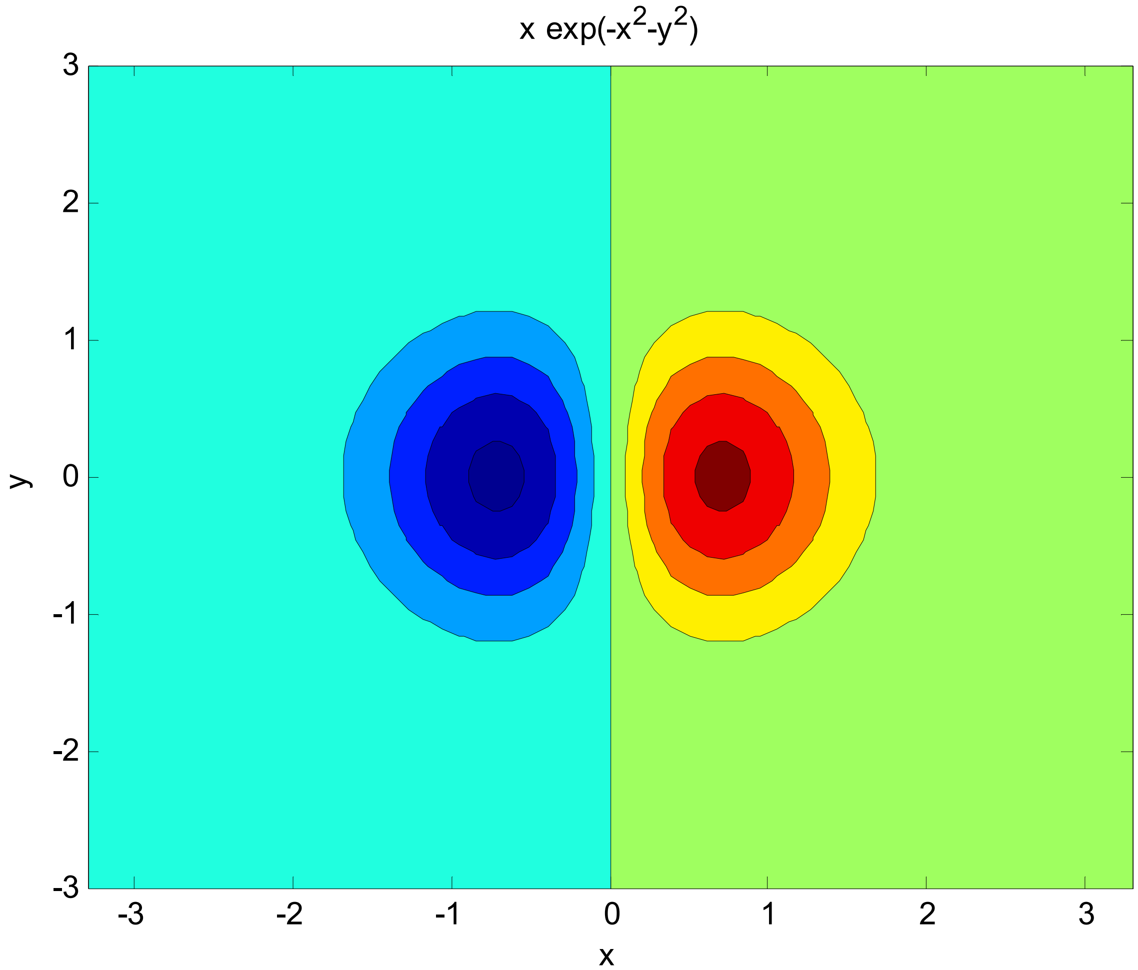


```
figure;
ezcontour('x*sin(y)',[-5 5 -5 5]);
```

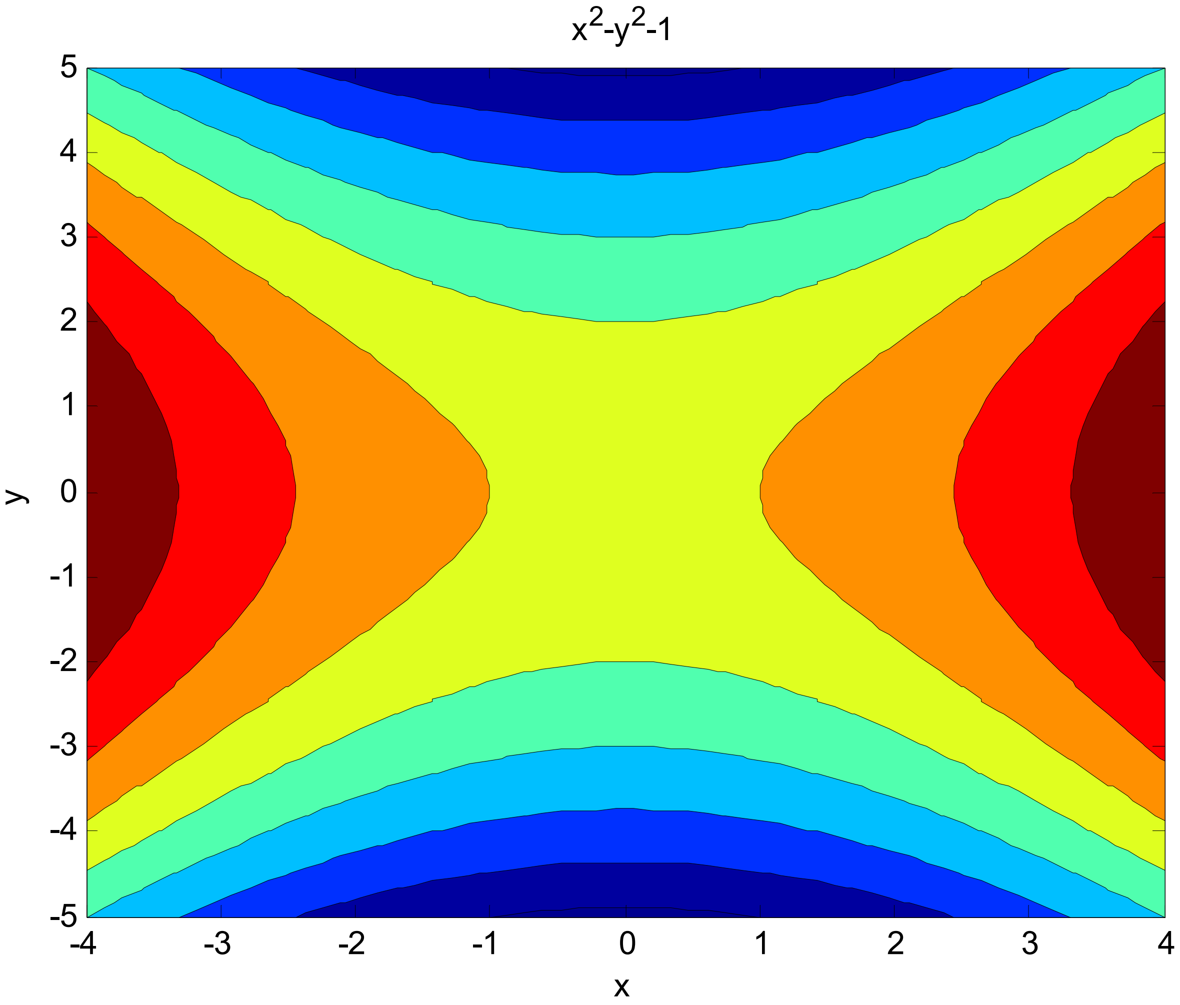


## ➢ 函数ezcontourf：绘制经过填充的符号函数等值线图

```
figure;
ezcontourf('x*exp(-x.^2-y.^2)');
```



```
figure;
ezcontourf('x^2-y^2-1',[-4 4 -5 5]);
```



## 6.9.4 符号函数的三维彩色曲面图

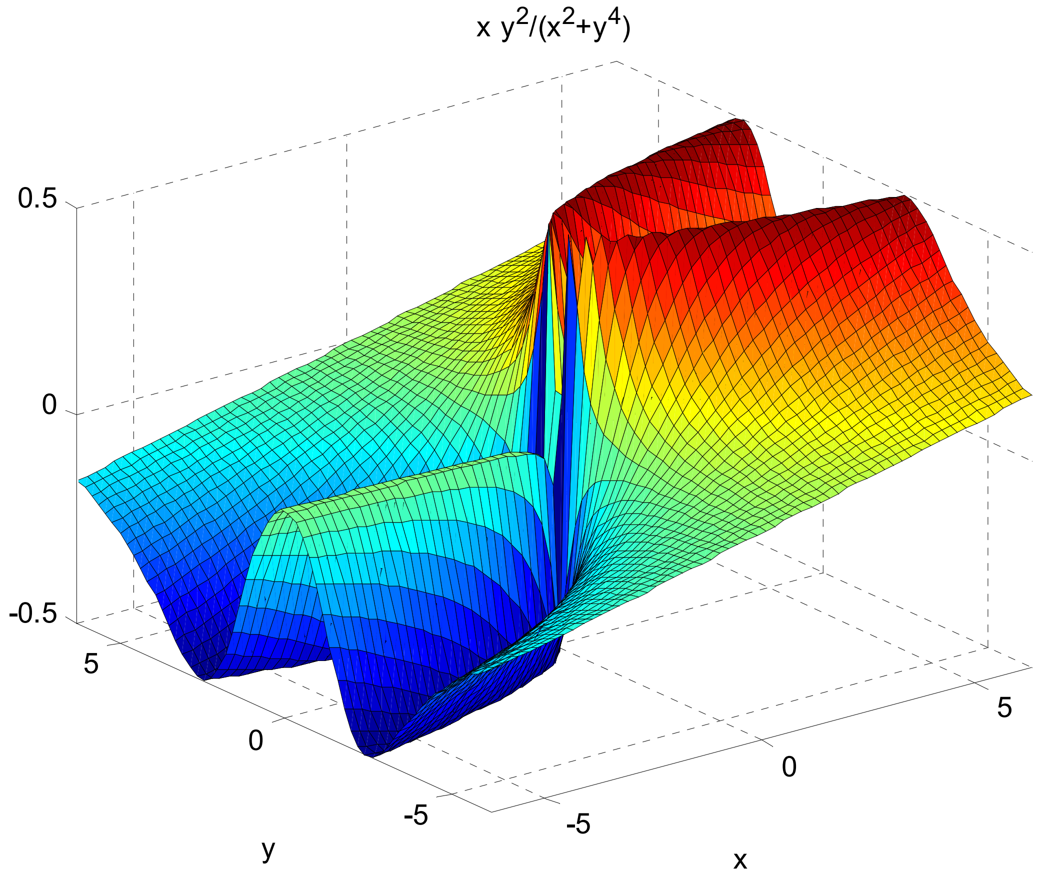
在MATLAB中，采用函数ezsurf( )绘制三维彩色曲面图，利用函数ezsurfc( )绘制带有等值线的三维彩色曲面图。

1．函数ezsurf( )

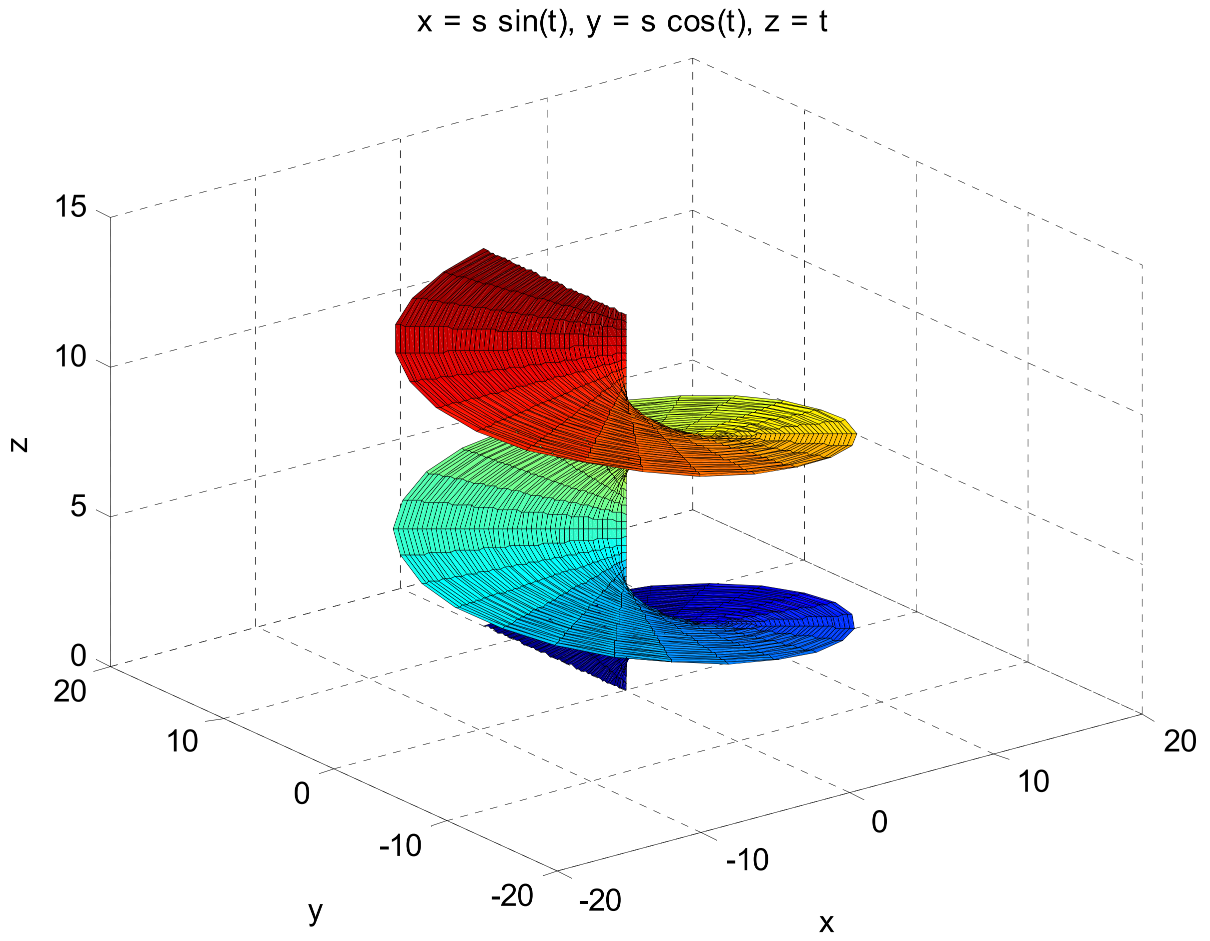
2．函数ezsurfc( )

## ➢ 函数ezsurf( )：绘制符号函数的三维彩色曲面图

```
figure;
ezsurf('x.*y.^2/(x.^2+y.^4)');
```

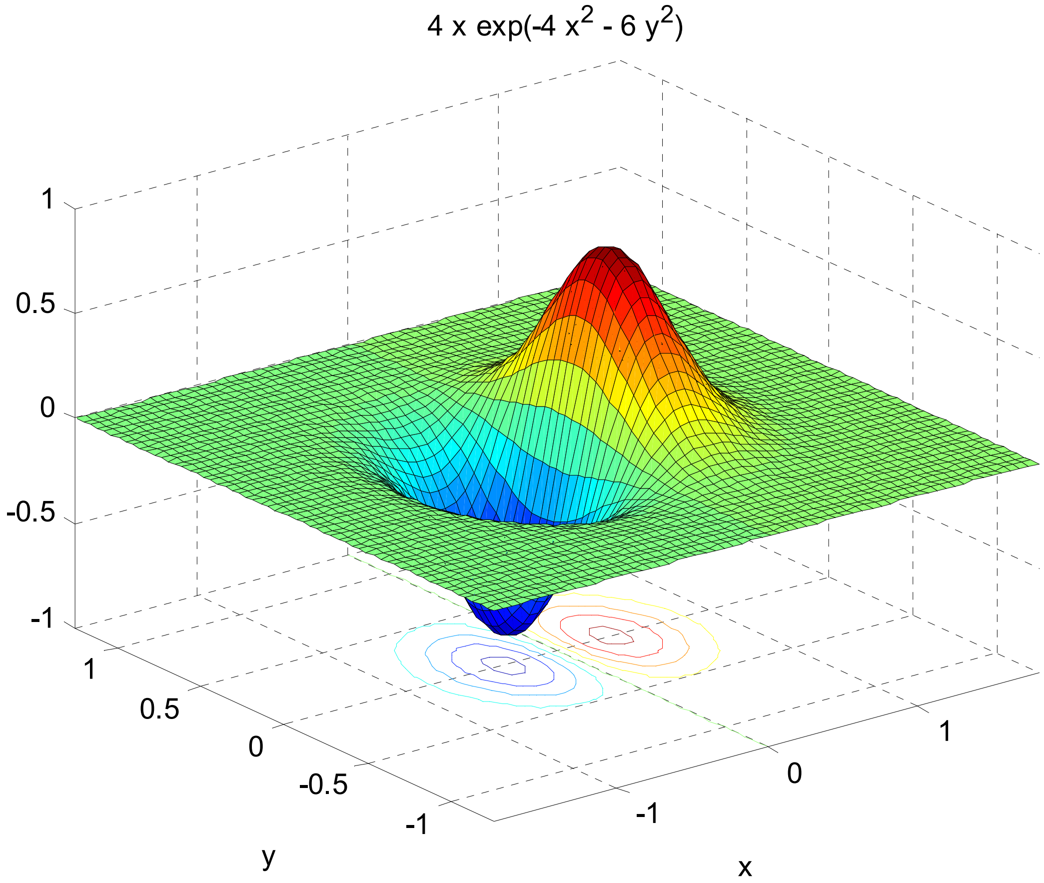


```
figure;
ezsurf('s*sin(t)','s*cos(t)','t',[0,4*pi]);
```

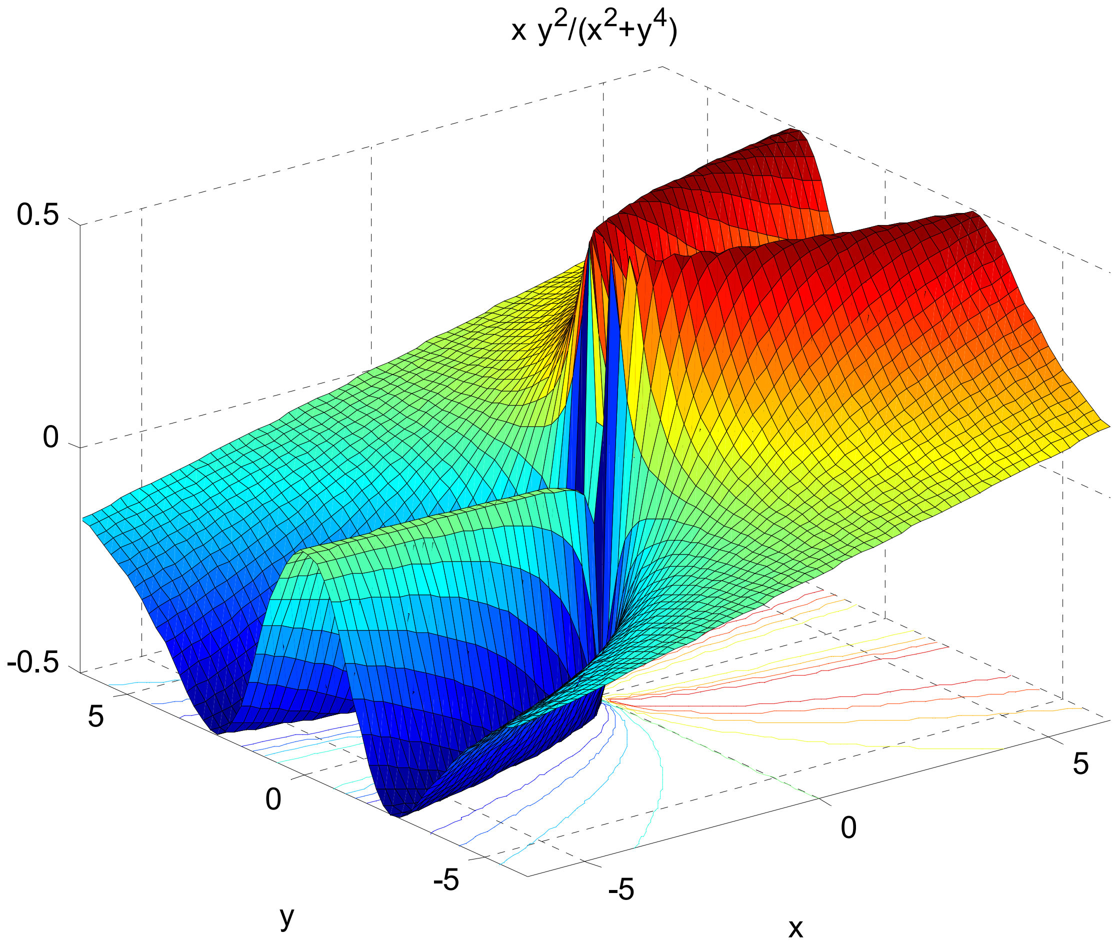


## ➢ 函数ezsurfc( )：绘制带有等值线的符号函数三维彩色曲面图

```
figure;
ezsurfc('4*x.*exp(-4*x.^2 - 6*y.^2)');
```



```
figure;
ezsurfc('x.*y.^2/(x.^2+y.^4)');
```



## 6.9.5 图形化符号函数计算器

在MATLAB中，提供了图形化的符号函数计算器。图形化的符号函数计算器功能虽然不是十分的强大，但是操作非常简单和方便，用户可以可以对符号运算和函数图形有个直观的了解。在MATLAB中，常用的有两种符号函数计算器，单变量符号函数计算器和泰勒级数逼近计算器。

## 单变量符号函数计算器

在MATLAB中，可以使用函数funtool来调用图形化的单变量符号函数计算器。

用户在命令行窗口中输入：funtool，可将单变量符号函数计算器调出，包括3个图形窗口。

# 泰勒级数逼近计算器

在MATLAB中，可以使用函数taylortool来调用图形化的泰勒级数逼近计算器。用户在命令行窗口中输入：taylortool，即可将泰勒级数逼近计算器调出。泰勒级数逼近计算器是一个交互的界面，可以绘制函数f(x)前N阶的泰勒级数。

系统默认的函数为f(x)=x*cos(x)，泰勒级数展开的阶数为7，自变量的默认区间为[-2*pi, 2*pi]。