

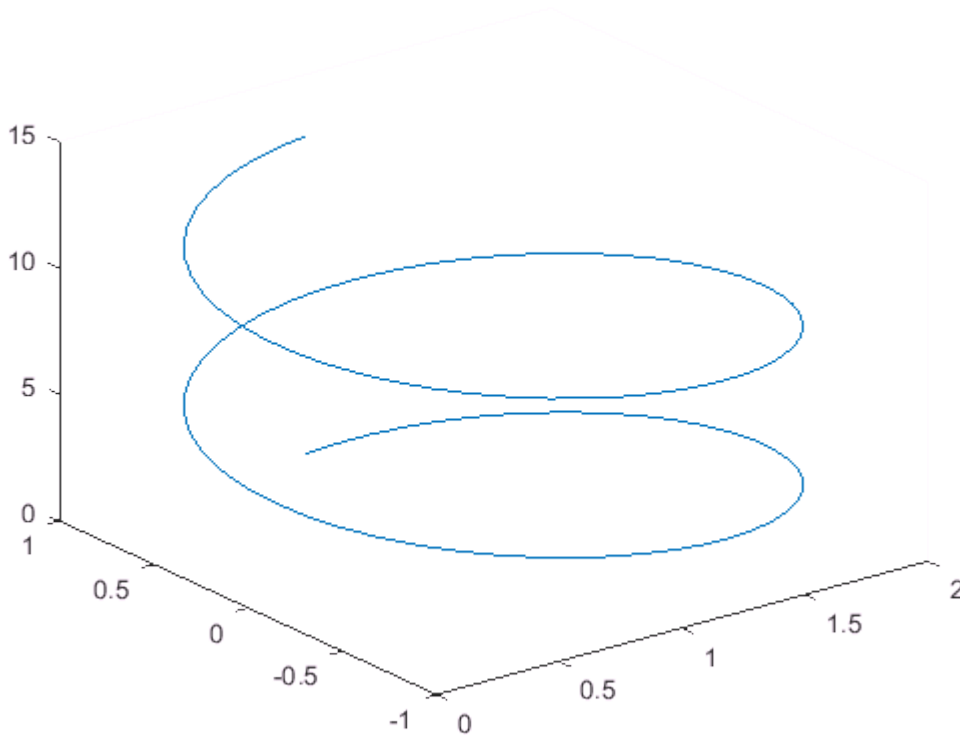
符号函数的三维图形绘制

```
clear;clc
```

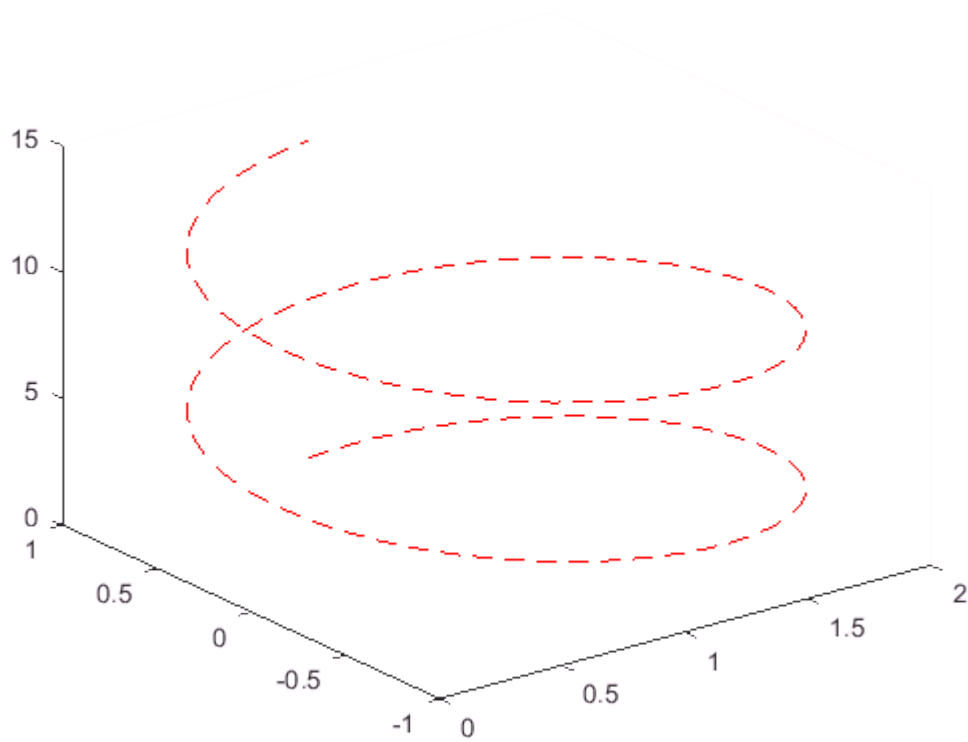
plot3函数 (类似于**plot**函数，实际上可以认为画的是三维空间下的折线图)

`t = 0:0.1:4*pi;` % 符号是t, t从0变动到4pi, 间隔设置为0.1

```
t = linspace(0,4*pi,100); % 符号是t, t从0变动到4pi, 等分为100个点  
x = sin(t)+1; % 利用t计算x的值  
y = cos(t); % 利用t计算y的值  
z = t; % 利用t计算z的值  
plot3(x,y,z) % 绘制三维空间下的折线图
```



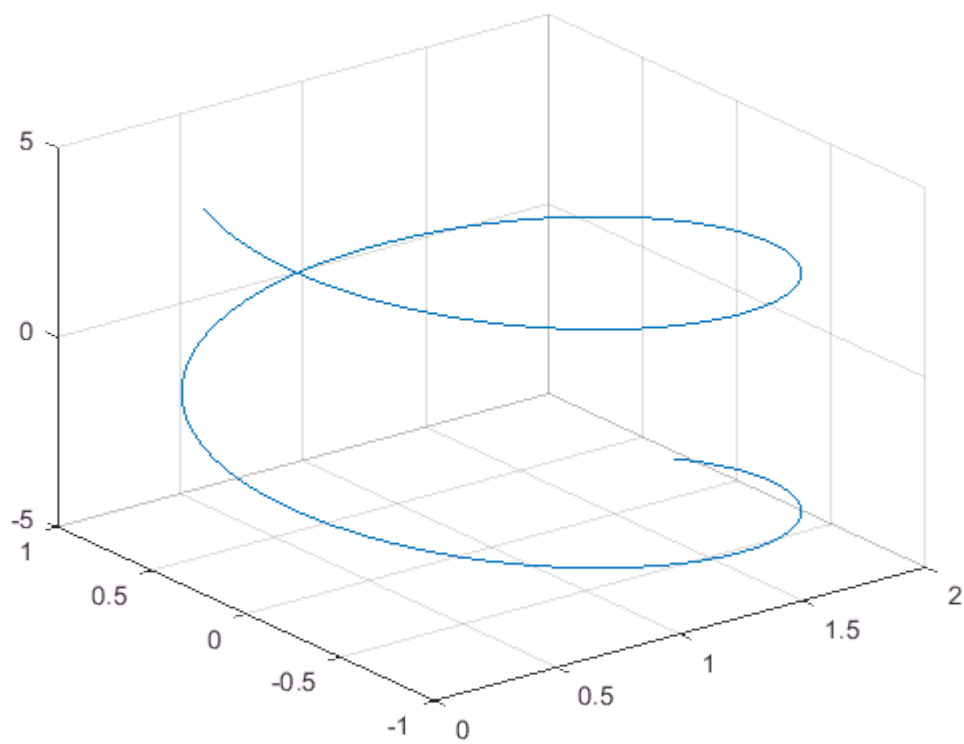
```
plot3(x,y,z,'r--')
```



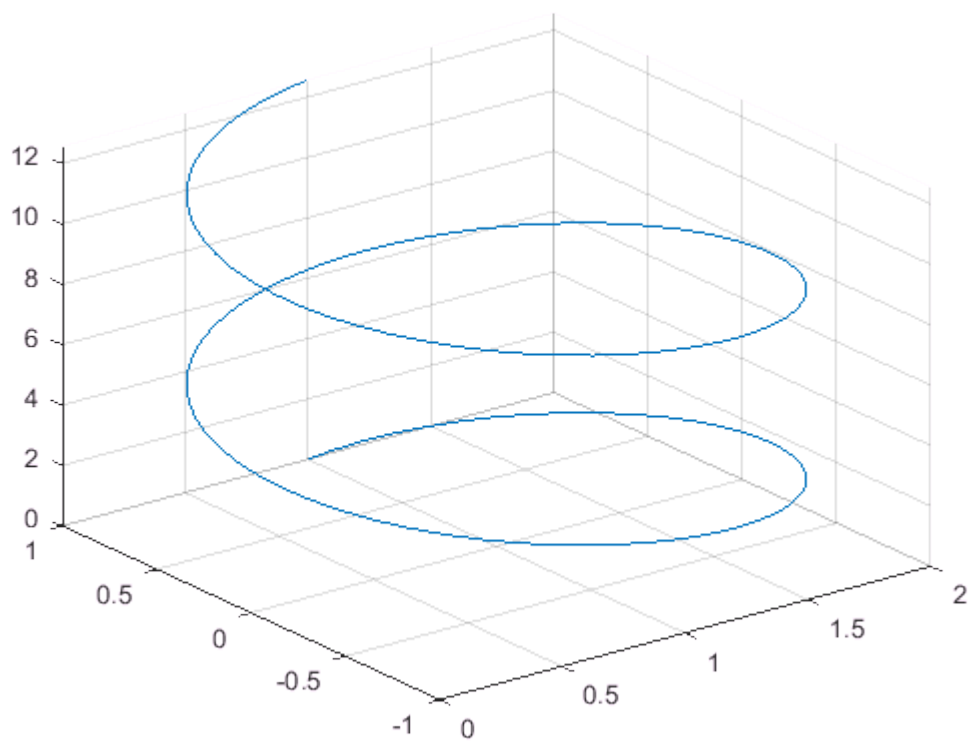
```
% close % 关闭图形
```

fplot3函数 (2016a版本推出的函数，老版本Matlab用的是**ezplot3**函数，未来版本可能会被淘汰)

```
syms t % 符号是t  
x = sin(t)+1;  
y = cos(t);  
z = t;  
fplot3(x,y,z) % 默认t的变化范围为[-5 5]
```



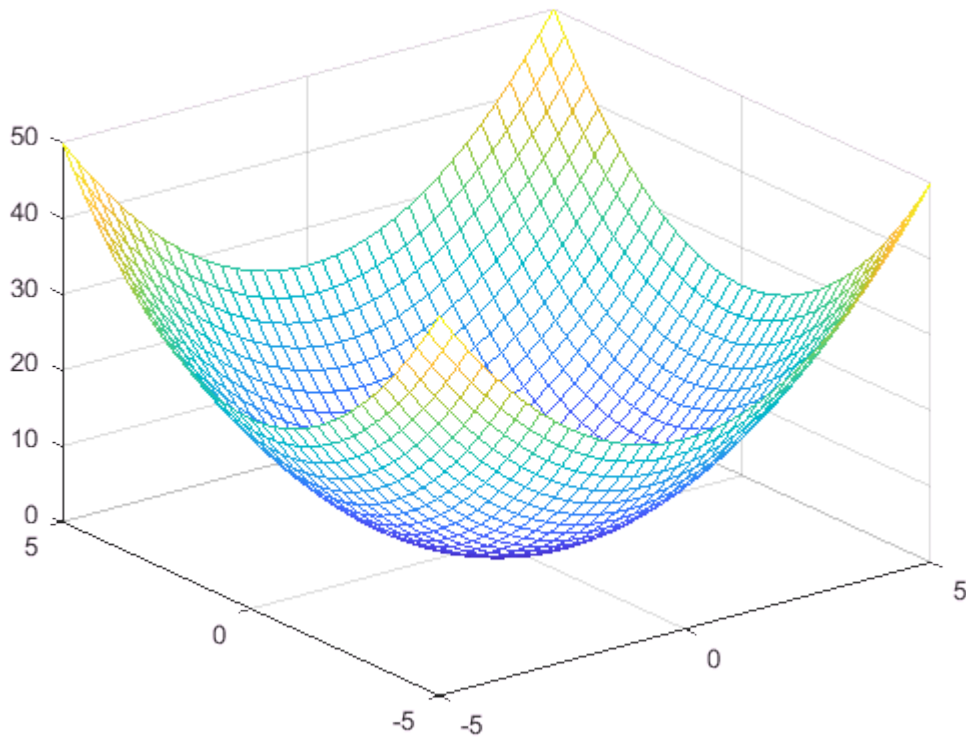
```
fplot3(x,y,z,[0 4*pi]) % 修改t的变化范围为0至4pi
```



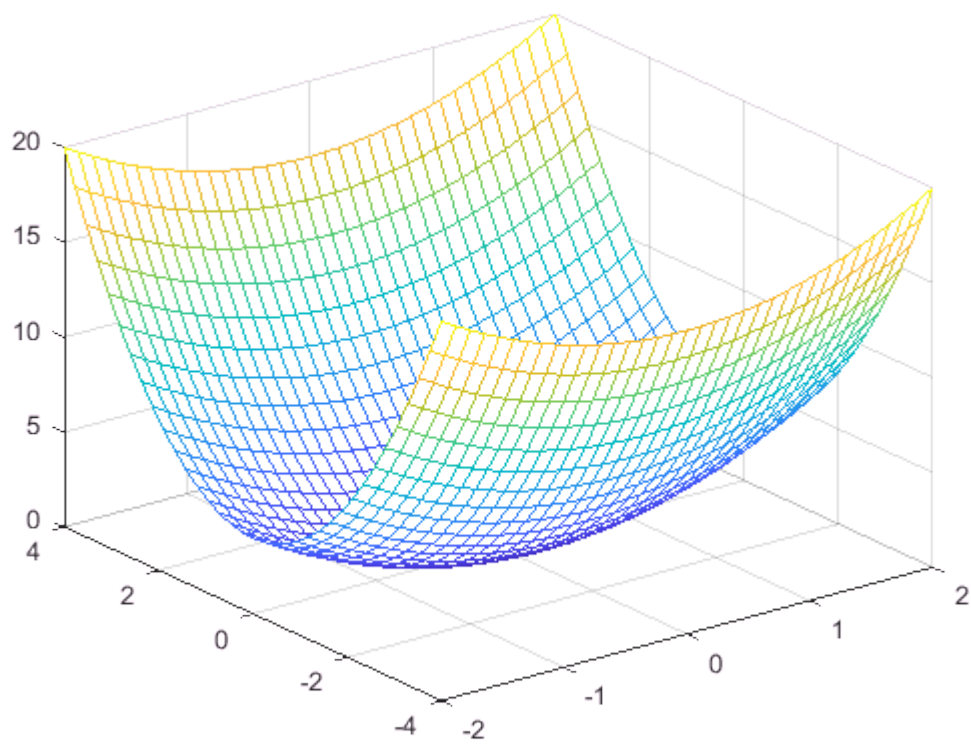
```
% close % 关闭图形
```

fmesh 三维网格图 (2016a版本推出的函数，老版本Matlab用的是ezmesh函数，未来版本可能会被淘汰)

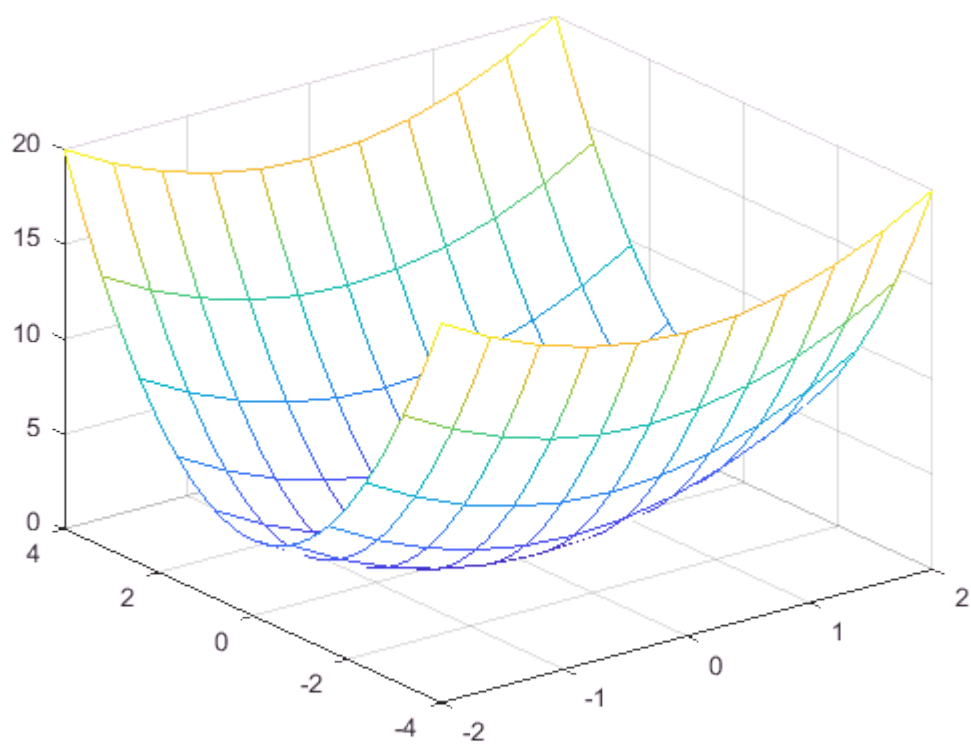
```
syms x y  
z=x^2+y^2;  
fmesh(z) % 默认x的变化范围和y的变化范围都是[-5 5]
```



```
fmesh(z, [-2 2 -4 4]) % 修改x的变化范围和y的变化范围分别为[-2 2]和[-4 4]
```



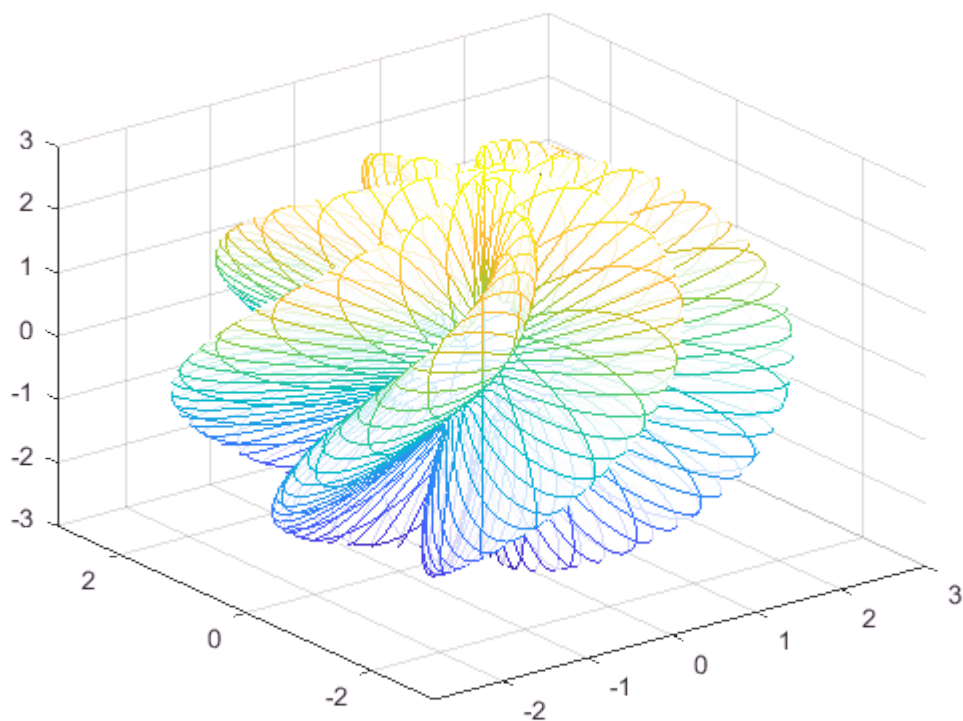
```
% axis vis3d % 冻结屏幕高宽比，使得一个三维对象的旋转不会改变坐标轴的刻度显示
% axis equal % 设置屏幕高宽比，使得每个坐标轴的具有均匀的刻度间隔
fmesh(z,[-2 2 -4 4], 'MeshDensity',11) % 设置每个方向计算的点数
```



```

% 画一朵花花送给亲爱的Ta
syms u v
r = 2 + sin(7.*u + 5.*v);
x = r.*cos(u).*sin(v);
y = r.*sin(u).*sin(v);
z = r.*cos(v);
% 可以直接用乘号
% r = 2 + sin(7*u + 5*v);
% x = r*cos(u)*sin(v);
% y = r*sin(u)*sin(v);
% z = r*cos(v);
fmesh(x,y,z,[0 2*pi 0 pi])

```



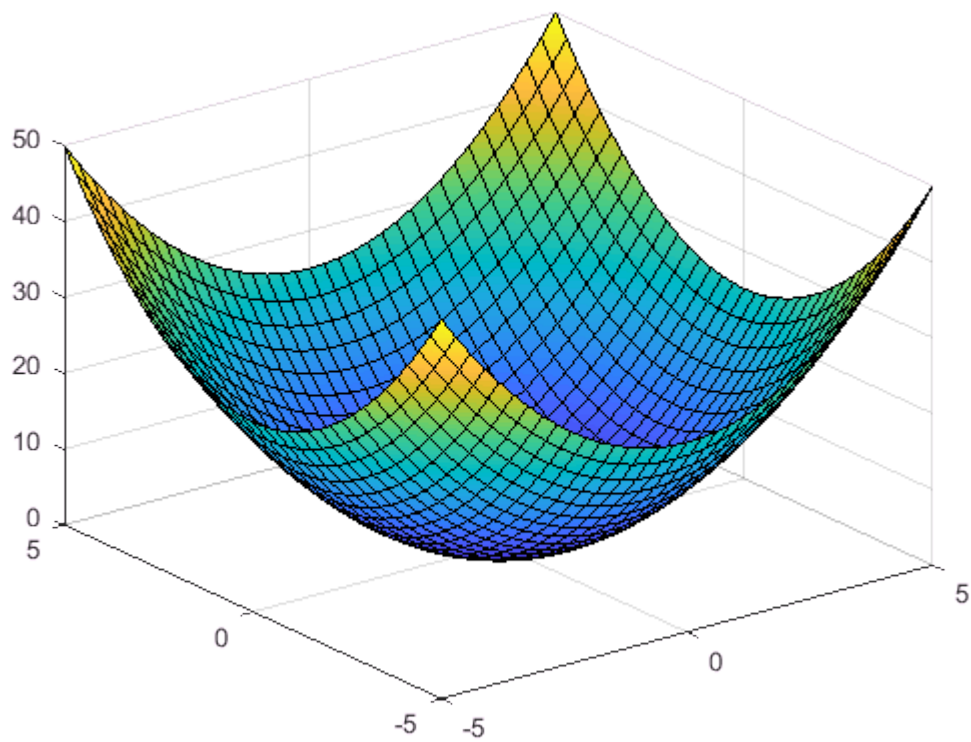
```
alpha(0.8)
```

fsurf 三维曲面图 (2016a版本推出的函数，老版本Matlab用的是ezsurf函数，未来版本可能会被淘汰)

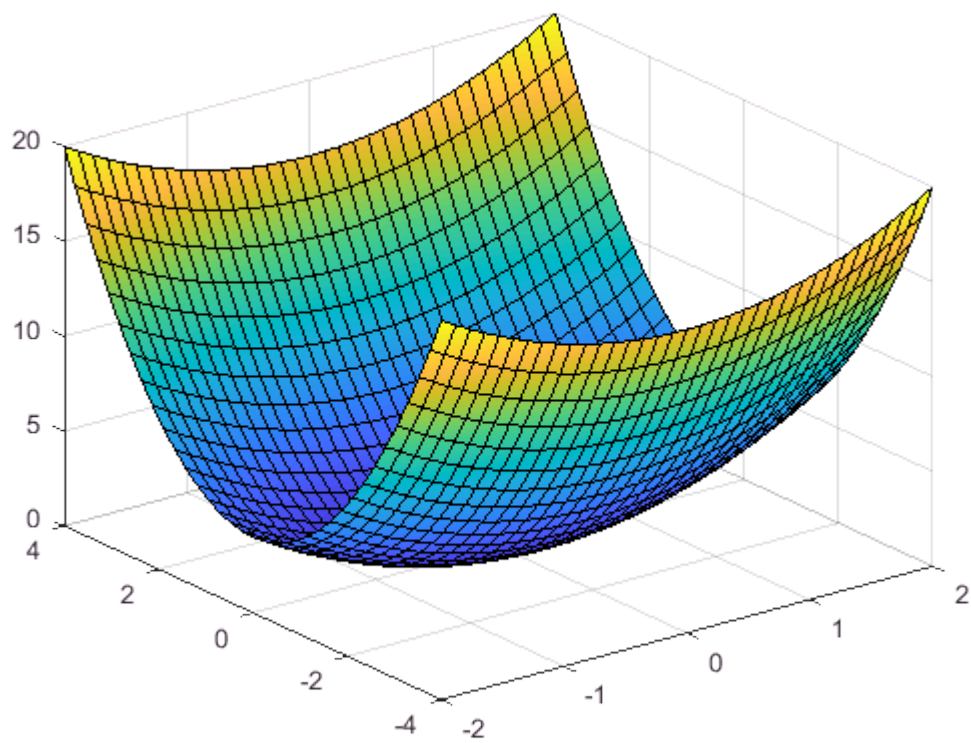
```

syms x y
z=x^2+y^2;
fsurf(z) % 默认x的变化范围和y的变化范围都是[-5 5]

```

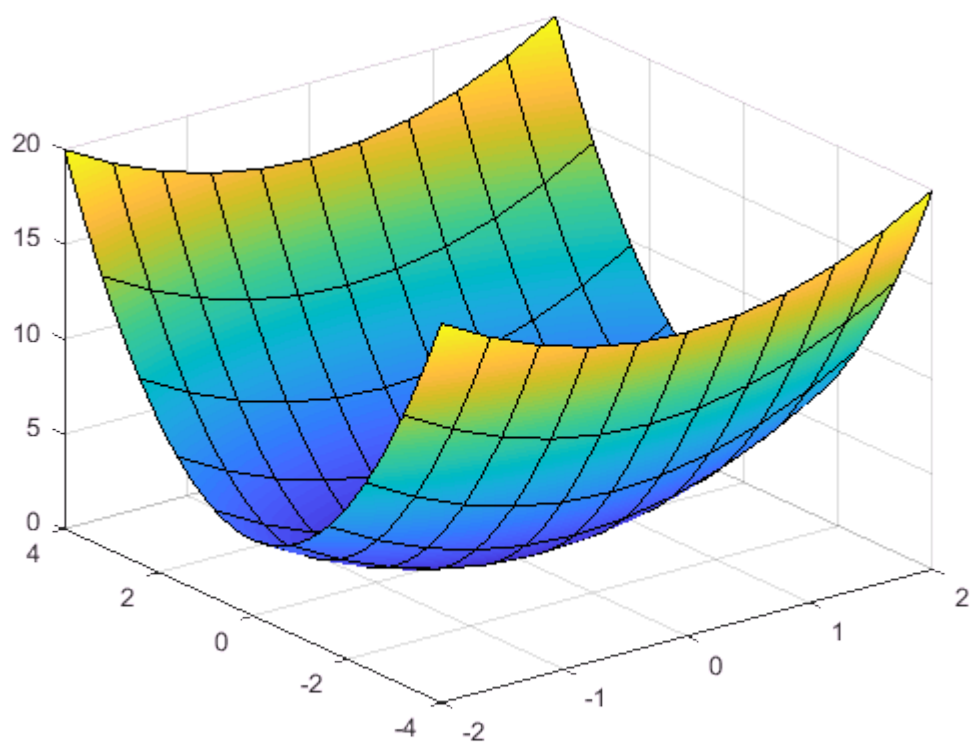


```
fsurf(z,[-2 2 -4 4]) % 修改x的变化范围和y的变化范围分别为[-2 2]和[-4 4]
```

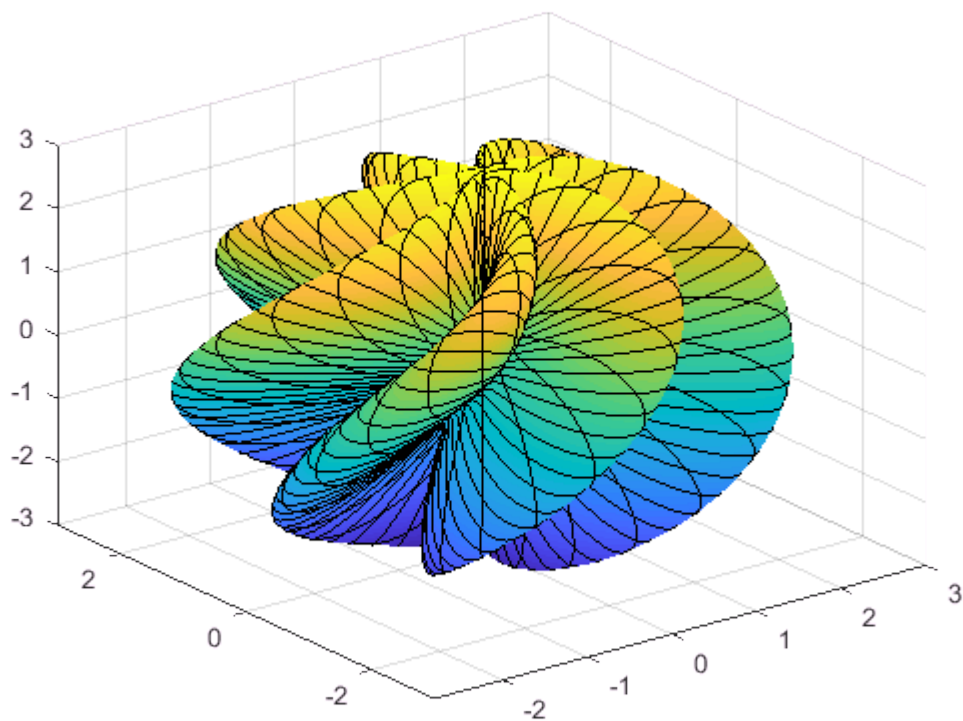


```
% axis vis3d % 冻结屏幕高宽比,使得一个三维对象的旋转不会改变坐标轴的刻度显示
% axis equal % 设置屏幕高宽比,使得每个坐标轴的具有均匀的刻度间隔
```

```
fsurf(z,[-2 2 -4 4], 'MeshDensity',11) % 设置每个方向计算的点数
```

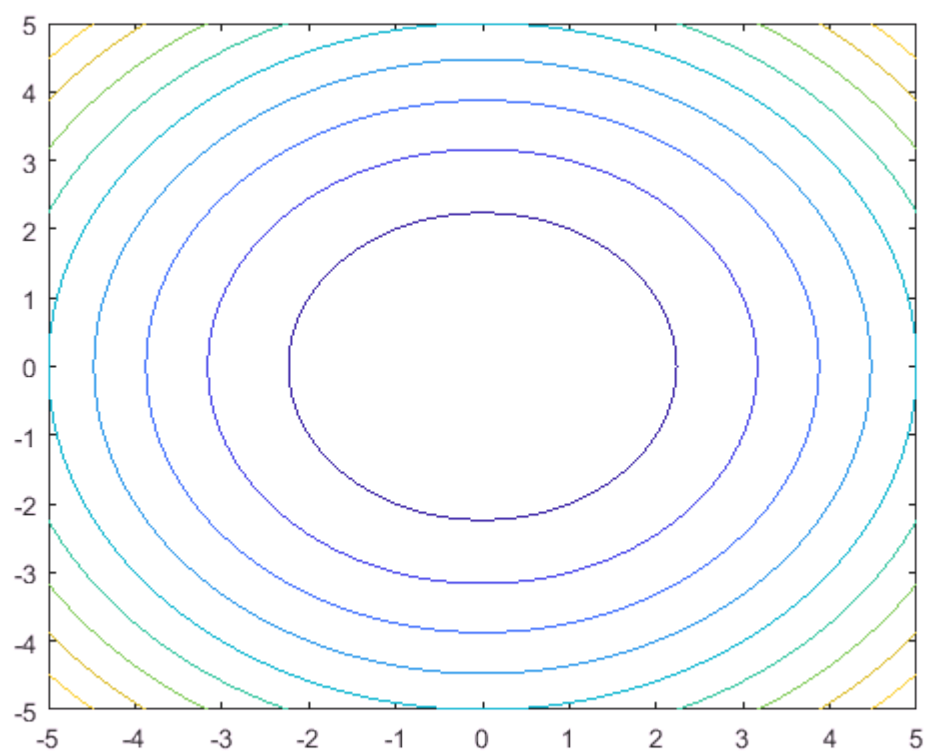


```
% 画一朵花花送给亲爱的Ta  
syms u v  
r = 2 + sin(7.*u + 5.*v);  
x = r.*cos(u).*sin(v);  
y = r.*sin(u).*sin(v);  
z = r.*cos(v);  
fsurf(x,y,z,[0 2*pi 0 pi])
```

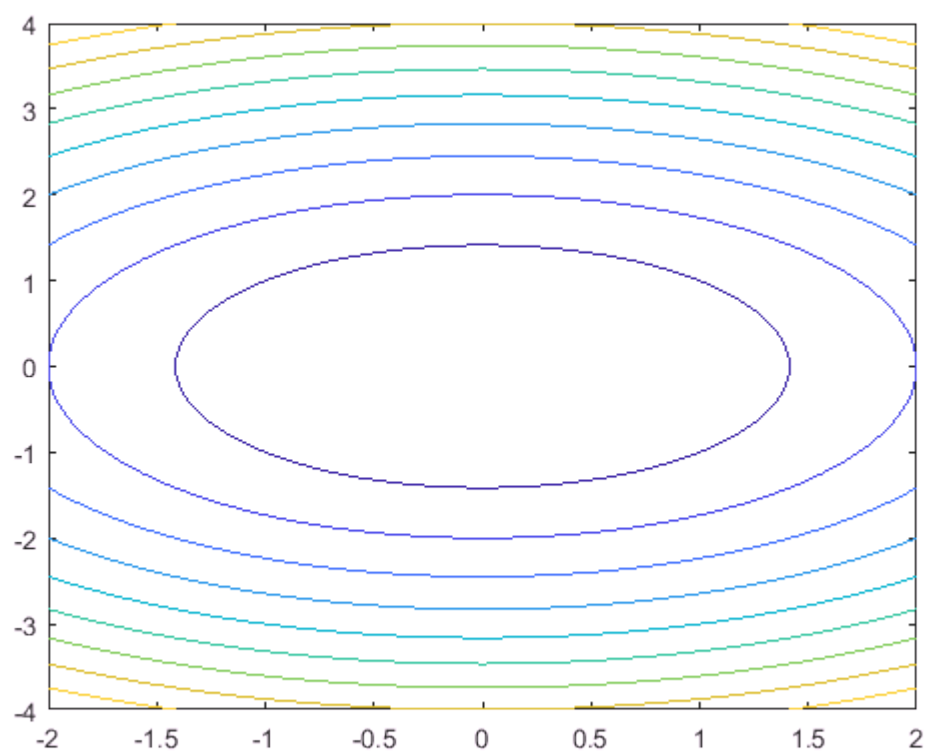



fcontour 绘制等高线 (2016a版本推出的函数，老版本Matlab用的是**ezcontour**函数，未来版本可能会被淘汰)

```
syms x y
z=x^2+y^2;
fcontour(z) % 默认x的变化范围和y的变化范围都是[-5 5]
```



```
fcontour(z,[-2 2 -4 4]) % 修改x的变化范围和y的变化范围分别为[-2 2]和[-4 4]
```



```
fcontour(z,'fill','on') % 在等高线条间进行填充
```

