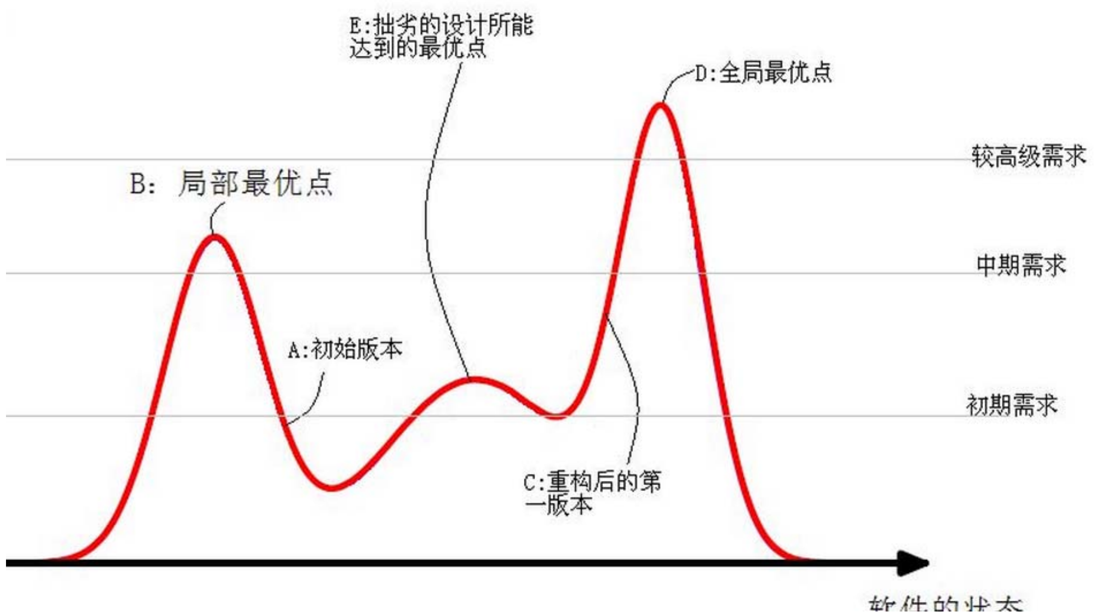
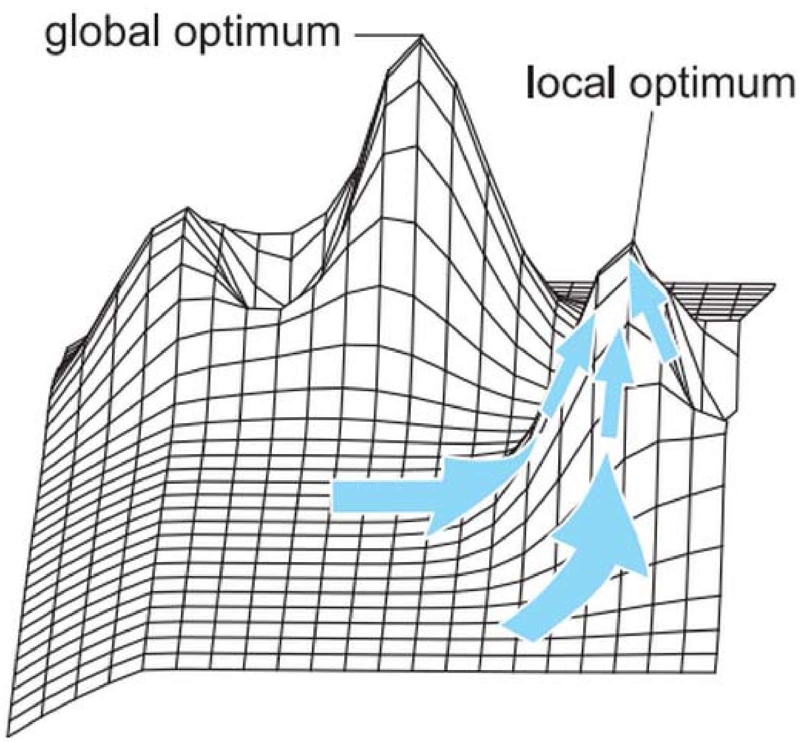
1、最优化就是模型得到最优解，也就是最优点和最优值。那么先要理解局部最优、全局最优。

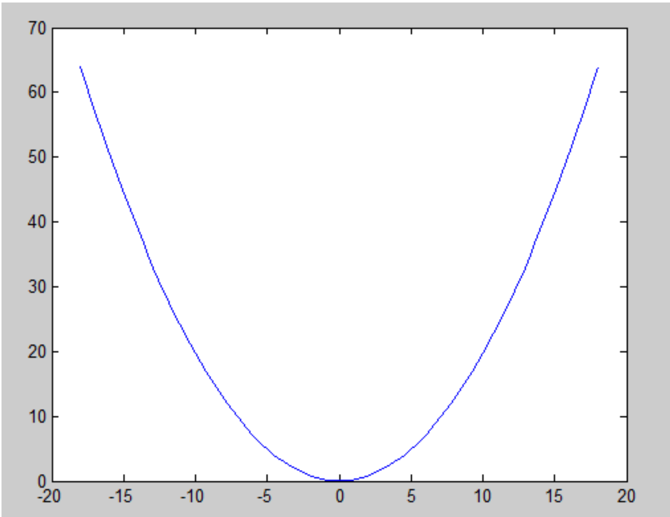




2、如何求解得到这个最优解。用导数、方向导数和梯度的概

念，同时需要借助 matlab 的图形功能（需要同学们自己加强）。

例子 1：y = x.^2，图形如下：



x=linspace(-18,18);%生成行向量，即生成横坐标的值

y=x.^2;%定义函数，得到纵坐标的值

plot(x,y);%画图，根据两个坐标值，画出图形

根据图形，可以看出最优解是 x = 0 ，可以用 matlab 根据导数的概念进行求解，如下：

syms x; %定义变量

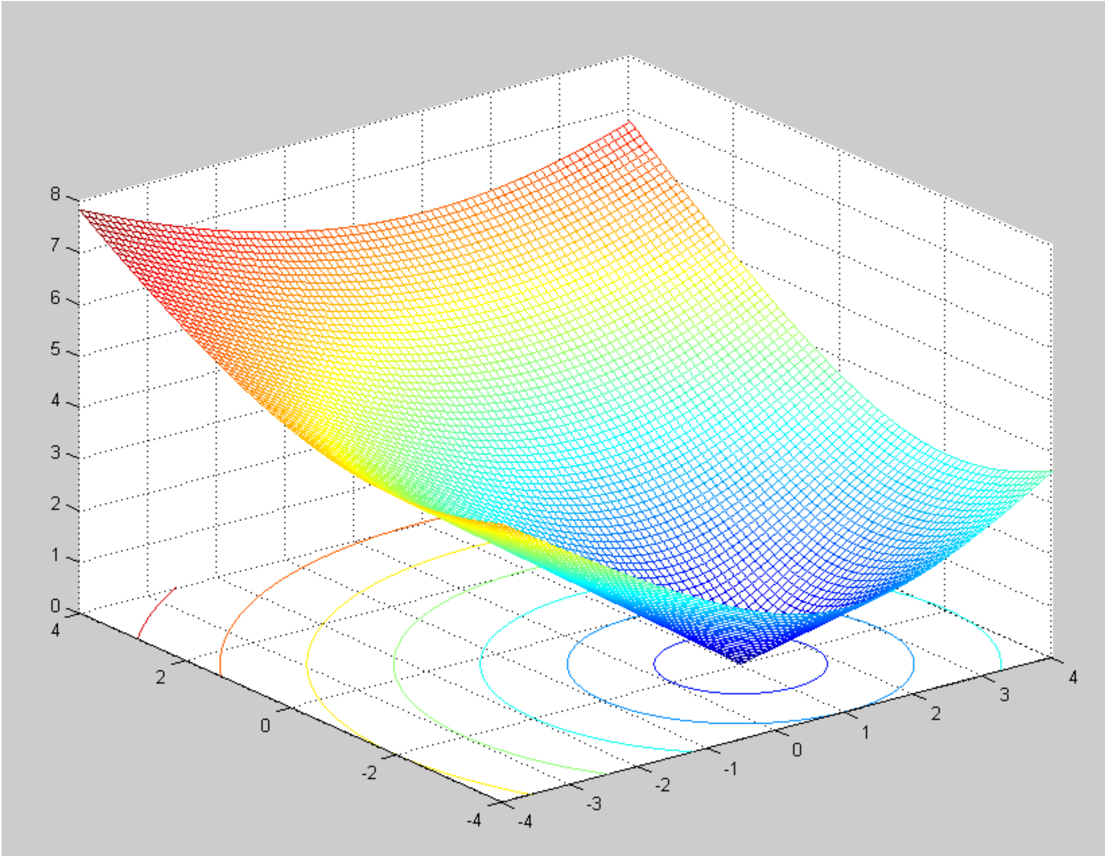
y=x.^2; %定义函数

s=diff(y); %求导 注：differential coefficient 导数

a=solve(s,'x'); %求根 或者直接 a=solve(s)？

b=double(a); %转换数值

例子 2：y =（x1-1）.^2+ （x2+2）.^2，图形如下：



[X1, X2] = meshgrid(-4:.1:4); %生成笛卡尔坐标网格

Y = (X1-1).^2 + (X2+2).^2; % Y = sqrt((X1-1).^2 + (X2+2).^2);

figure;%创建图形窗口，这样就可以生成很多个图形

meshc(X1,X2,Y);%生成 mesh 图和 contour 图（等高线）

利用导数的知识，求导得到最优解，如下：

syms y x1 x2; %定义变量

y=(x1-1).^2+(x2+2).^2; %定义函数

s=[diff(y,x1),diff(y,x2)];%分别求导

[Sx,Sy]=solve(s(1),s(2));%求根

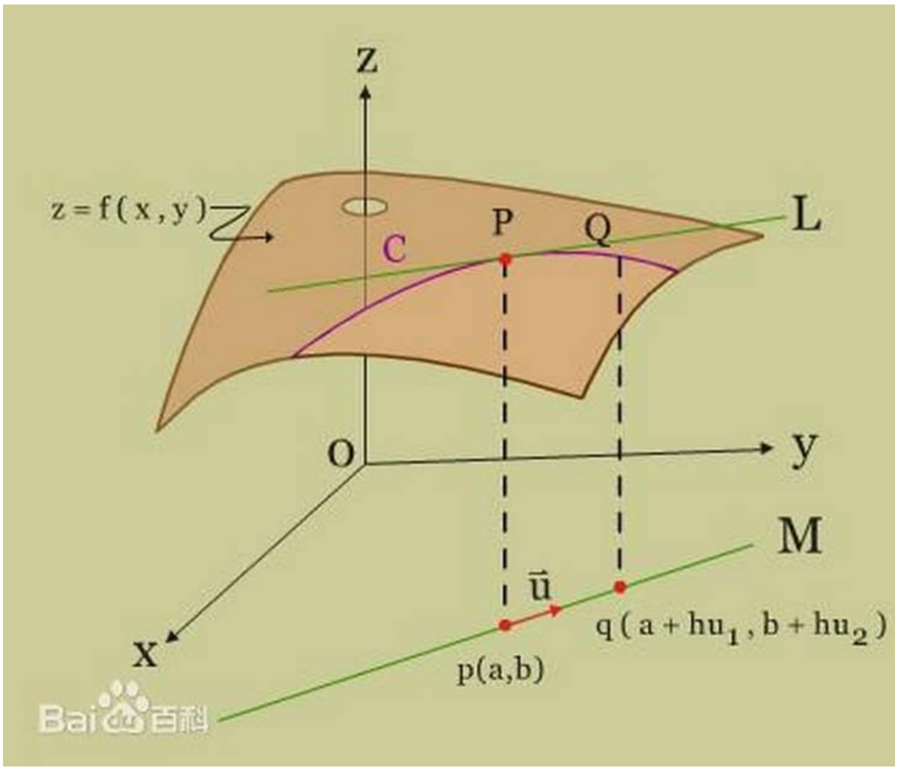
z=double([Sx,Sy])%数值化

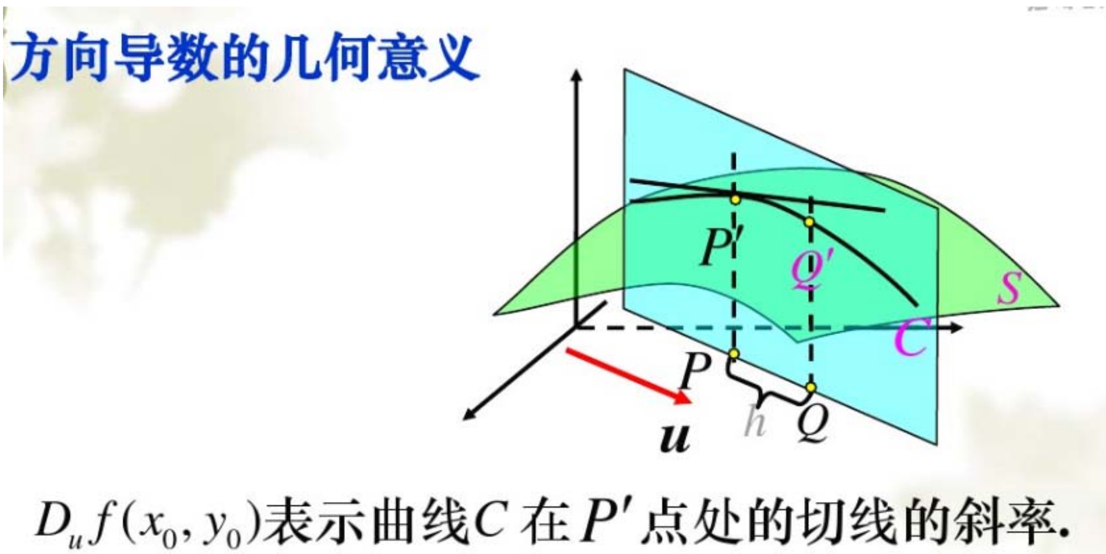
或者，采用梯度的函数，如下：

syms y x1 x2; %定义变量

```
y=(x1-1).^2+(x2+2).^2; %定义函数
grad_ff=gradient(y,[x1,x2]);%直接用梯度函数求解
[Sx,Sy]=solve(grad_ff);%求根
z=double([Sx,Sy]);%数值化
```

3、那什么是梯度呢？梯度的概念，先说一下方向导数的概念，如图：





梯度就是在 360 度的方向中，某一个方向的方向导数。

问题：函数在点 P 沿哪一方向增加的速度最快？

定义 设函数 $z=f(x,y)$ 在平面区域 D 内具有一阶连续偏导数，则对于每一点 $P(x,y)\in D$，都可定出一个向量 $\frac{\partial f}{\partial x}\vec{i}+\frac{\partial f}{\partial y}\vec{j}$，这向量称为函数 $z=f(x,y)$ 在点 $P(x,y)$ 的**梯度**，记为

$$\boxed{\overrightarrow{grad}\ f(x,y)=\frac{\partial f}{\partial x}\vec{i}+\frac{\partial f}{\partial y}\vec{j}.}$$

看例子：

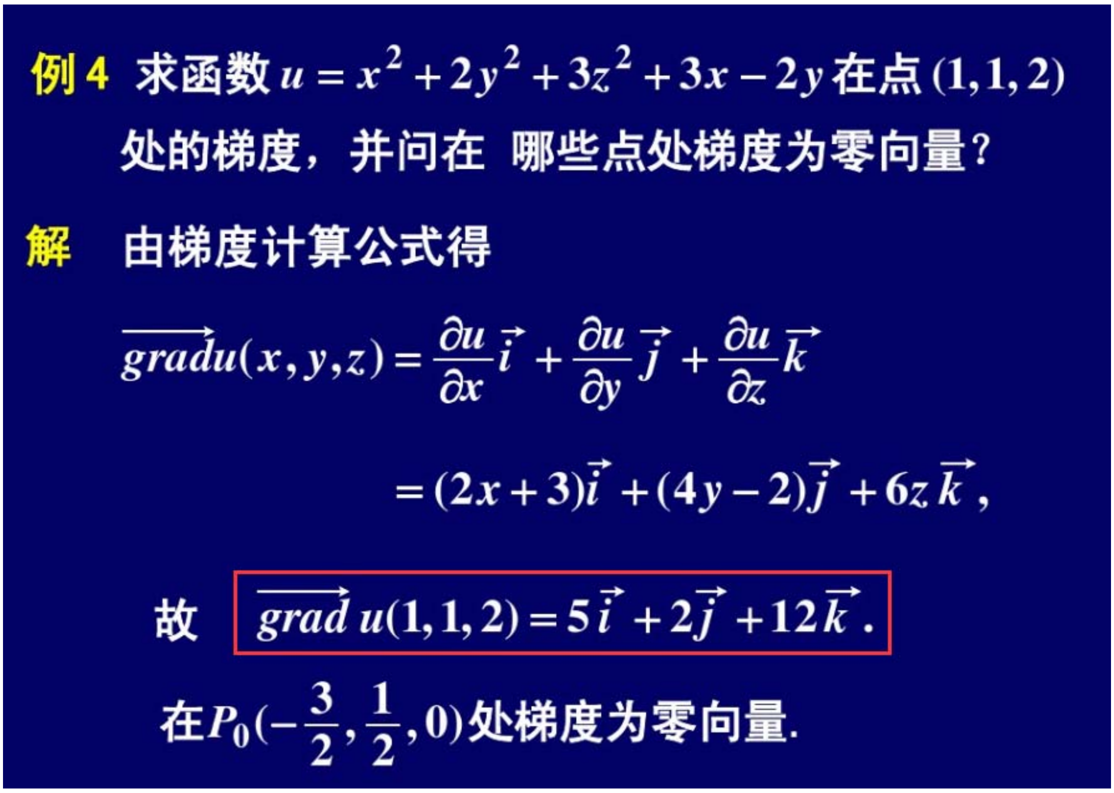
例 2 求函数 $f(x,y)=x^2-xy+y^2$ 在点（1，1）沿与 x 轴方向夹角为 α 的方向射线 $\vec{l}$ 的方向导数.并问在怎样的方向上此方向导数有

（1）最大值；（2）最小值；（3）等于零？

用 matalb 求解函数在某个具体点的梯度，代码如下：

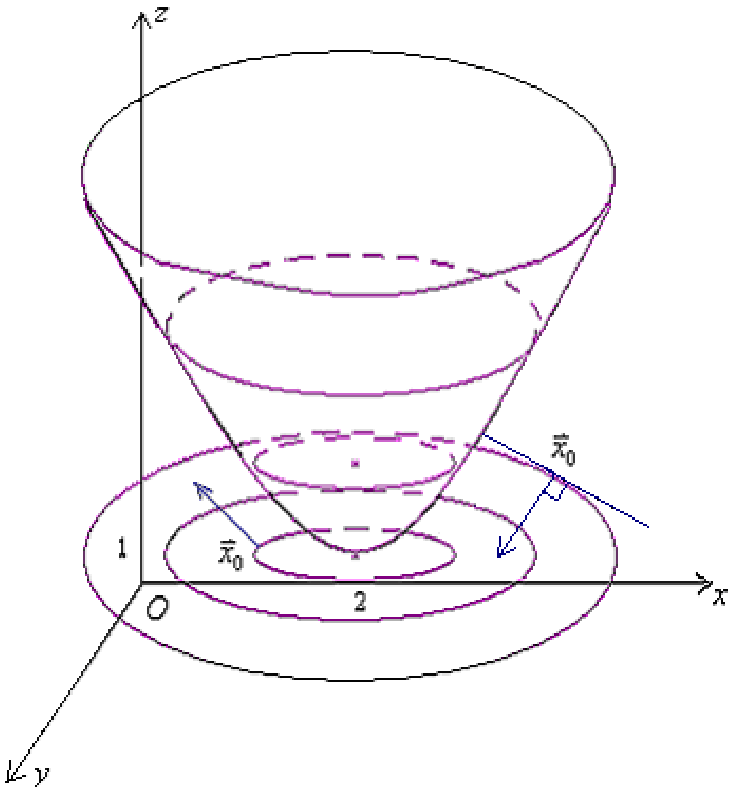
```
syms z x y; %定义变量
z=x.^2-x*y+y.^2; %定义函数
grad_ff=gradient(z,[x,y]);%直接用梯度函数求解
grad2 = subs(grad_ff,{x,y},{1,1})%用（1，1）求出具体值
```

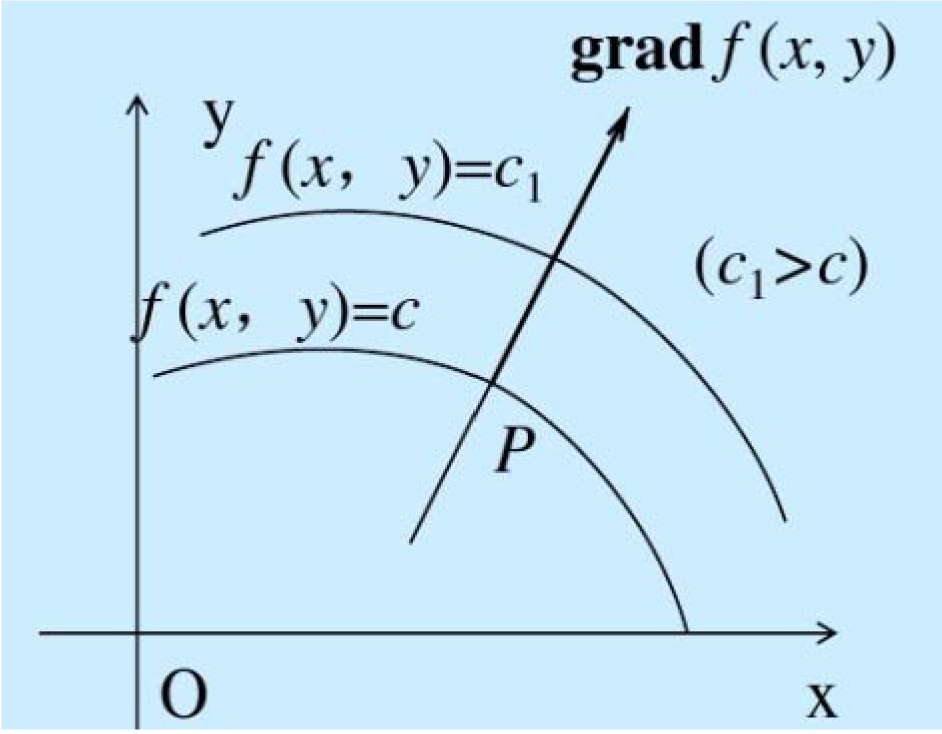
最后得到梯度为（1，1），那就是 45 度的方向。就是说，45 度的方向，导数的值最大，45 的反方向就是 45+180，是负梯度方向，与这个方向垂直的方向 45+90 和 45+270 是导数值为 0。



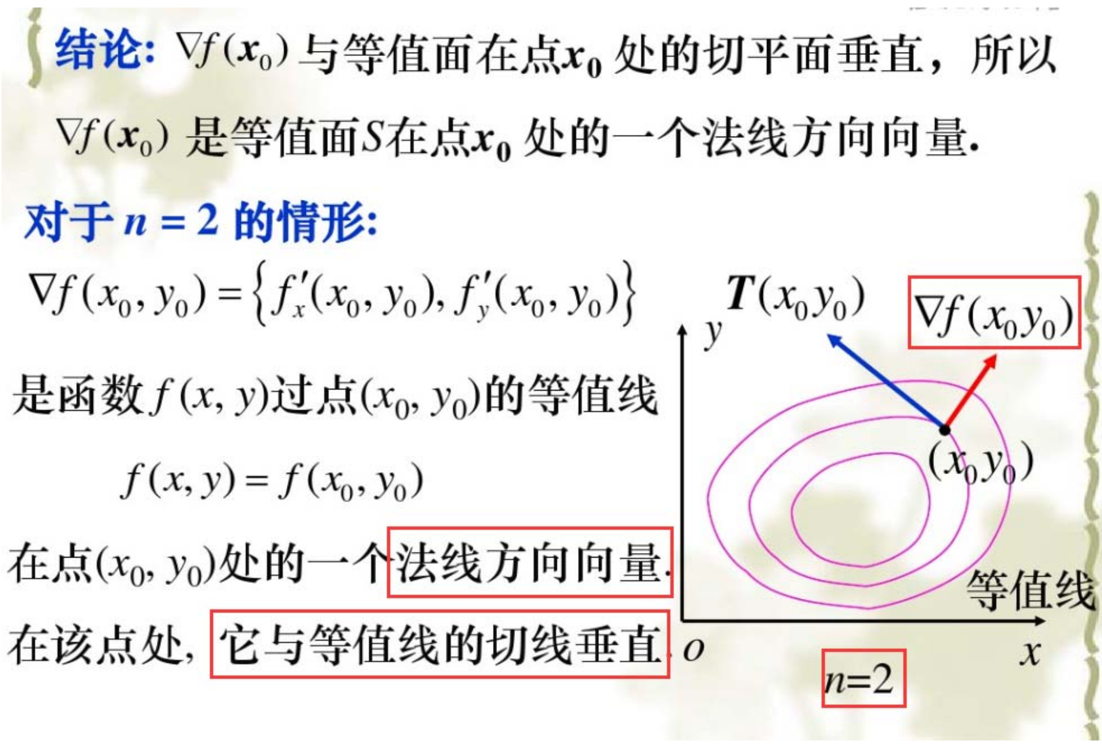
可以用上面的代码验证这道题对不对。

从上往下看，梯度和等高线的关系，如下：



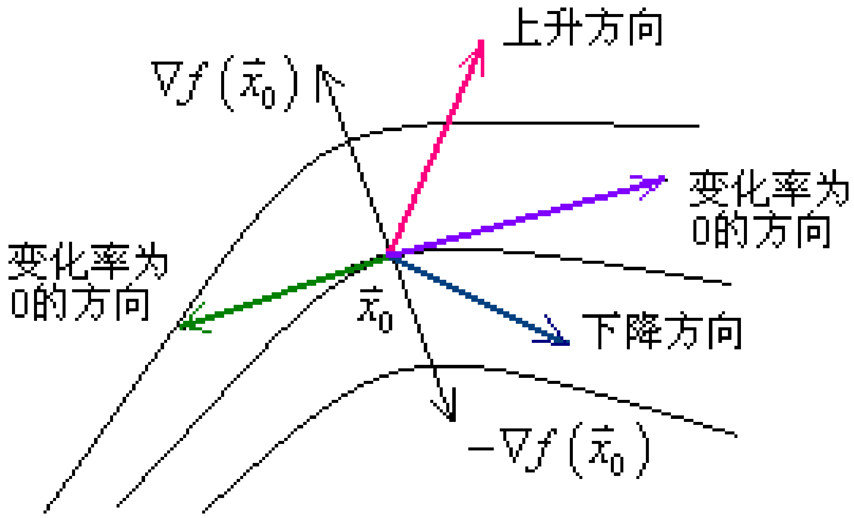


grad $f(x, y)$
y
$f(x, y)=c_1$
$(c_1>c)$
$f(x, y)=c$
P
O
x

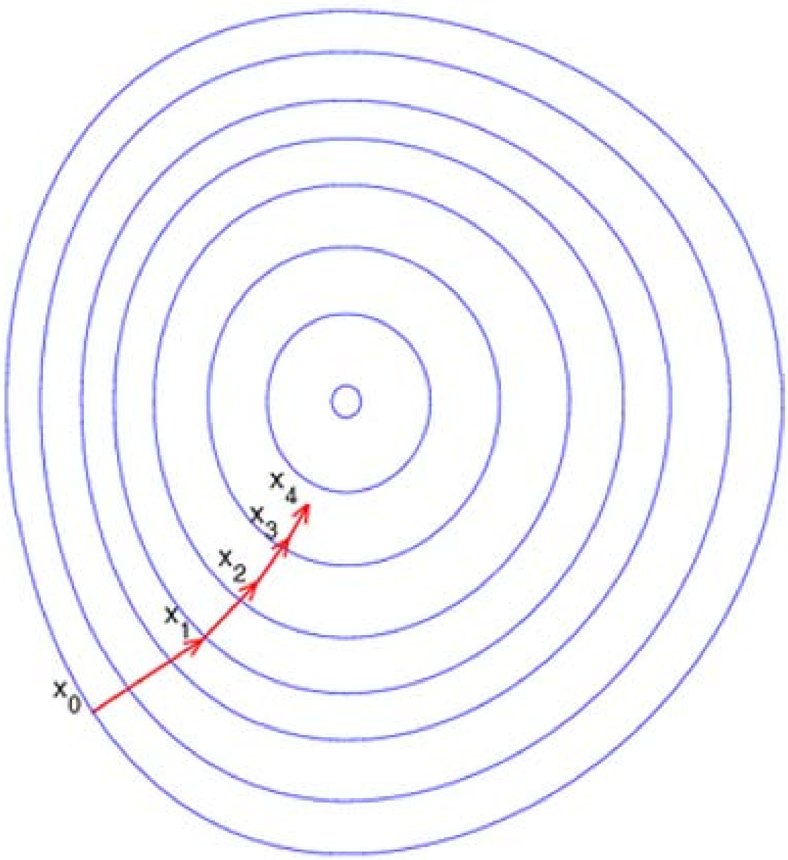


结论: $\nabla f(\boldsymbol{x}_0)$ 与等值面在点$\boldsymbol{x}_0$ 处的切平面垂直，所以
$\nabla f(\boldsymbol{x}_0)$ 是等值面S在点$\boldsymbol{x}_0$ 处的一个法线方向向量.
对于 $n=2$ 的情形:
$\nabla f(x_0, y_0)=\left\{f_x'(x_0, y_0), f_y'(x_0, y_0)\right\}$
是函数$f(x, y)$过点(x_0, y_0)的等值线
$f(x, y)=f(x_0, y_0)$
在点(x_0, y_0)处的一个法线方向向量.
在该点处，它与等值线的切线垂直.
$\boldsymbol{T}(x_0 y_0)$
$\nabla f(x_0 y_0)$
y
$(x_0 y_0)$
等值线
o
x
$n=2$

总结，看图：



4、除了用“梯度”的概念（梯度为 0）去求解最优值，还可以用一种“迭代”的思路求解最优解，或者说是“迭代算法”。看图如下：

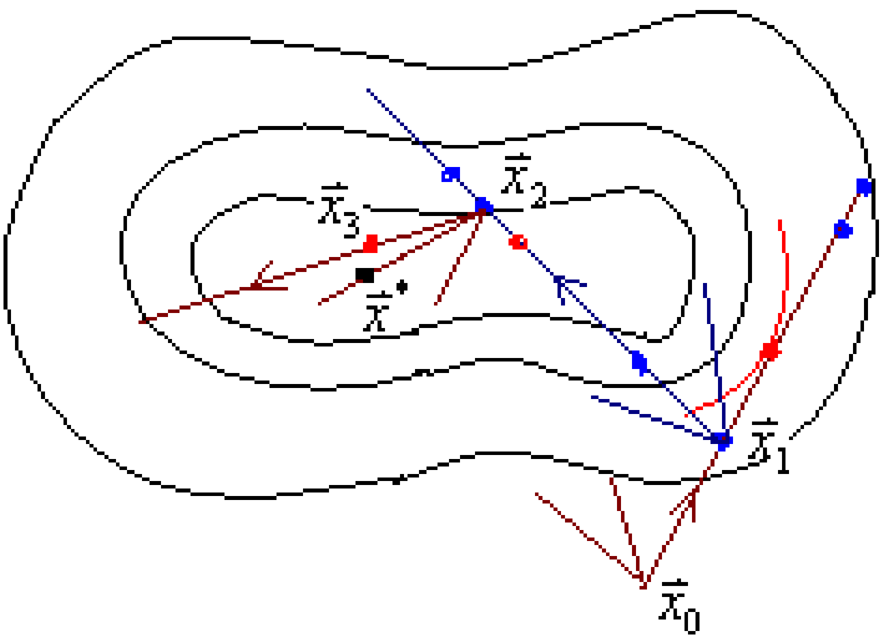


这种迭代的算法，我们可以叫做“下降迭代算法”。看图：

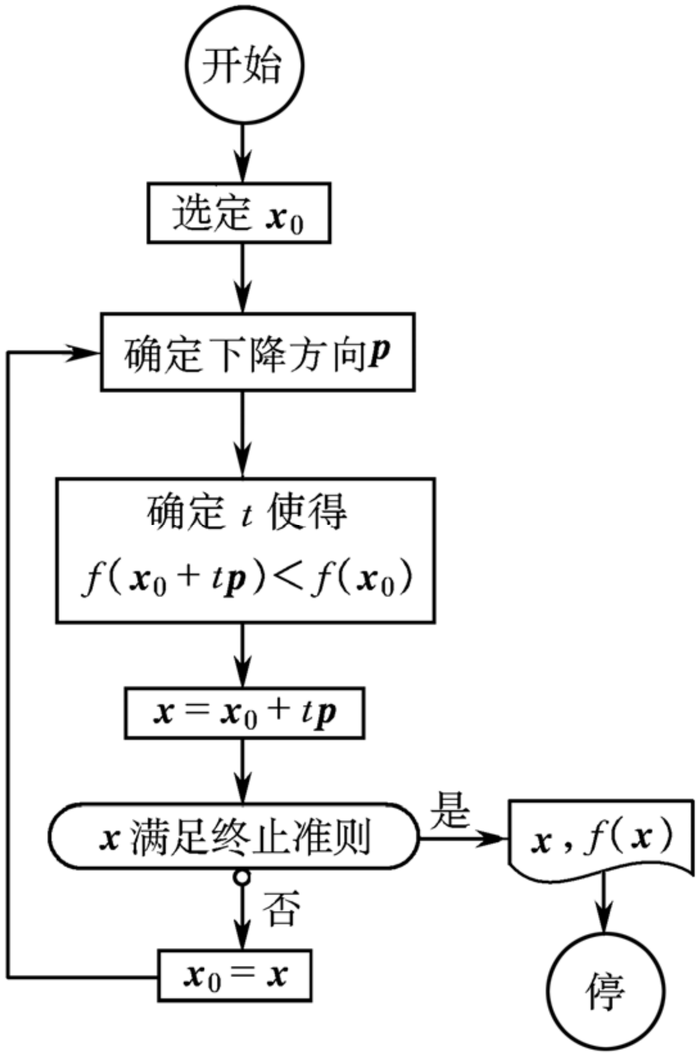
在集合 X 上的迭代算法是指：开始，选定一个初始点 $x_0 \in X$，置 $k=0$。然后按某种规则 A，把第 k 次迭代点 x_k 映射为新的一点 $x_{k+1} \in X$，记为 $x_{k+1} = A(x_k)$，并置 $k=k+1$. 这称为完成了第 $k+1$ 次迭代。这个过程无限地进行下去，就会产生一个点列 $\{x_k\}$。因此，规则 A 称为算法。

若点列 $\{x_k\}$ 收敛于 X，则称算法 A 在 X 上是收敛的；否则，称算法 A 在 X 上是发散的。特别地，当 X 是最优化问题（1.6）的容许集时，若除满足计算终止准则的迭代点外，对于每个 k，都有 $f(\check{x}_{k+1}) < f(\check{x}_k)$，则称 A 为下降迭代

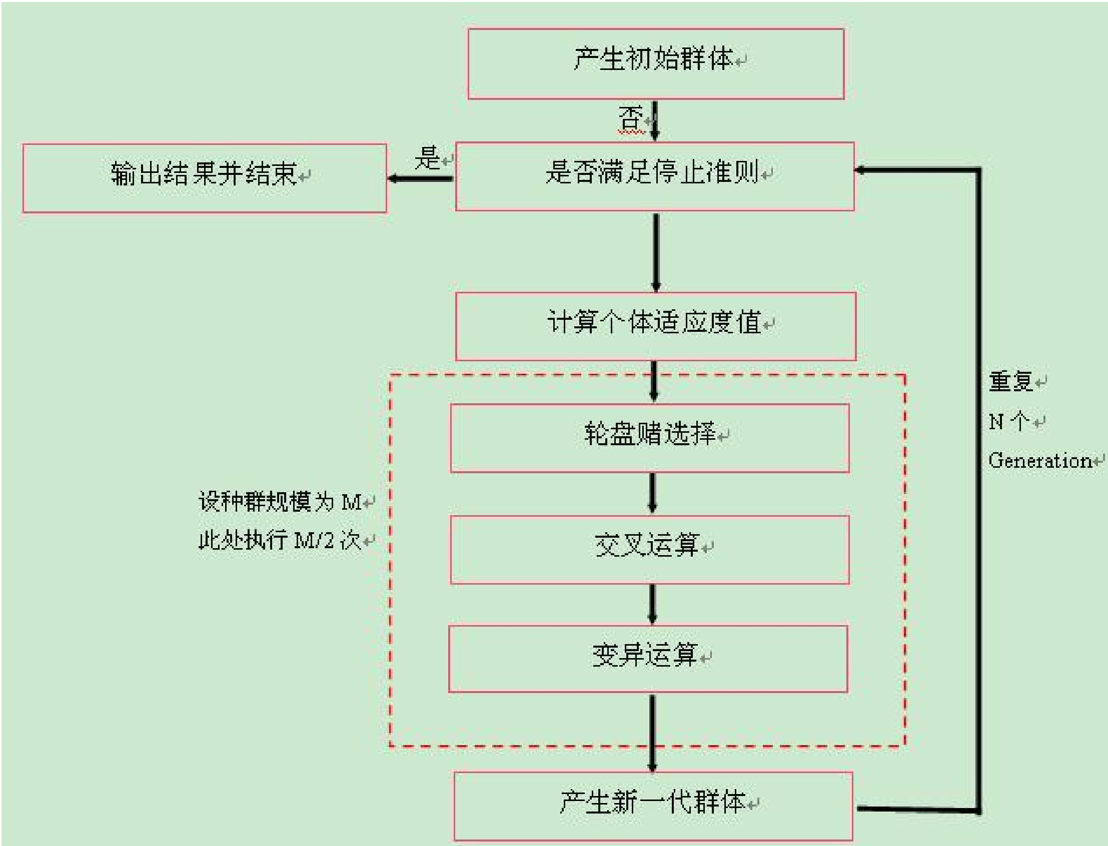
现在问题就是，最优化学习的主要内容就是这种算法，因此要学习不同的算法，可以让初始点迭代到最优点。其中有一种典型的指导规则，就是利用“梯度下降”的概念引导迭代，看图：



算法流程图为：



那还有一种指导规则，叫做“启发式规则”（会利用到概率），用某种知识去引导从“初始点”逼近“最优点”。这里面的典型算法就是 GA，利用进化的知识去模拟迭代算法，它们也是一种迭代算法，如下图。



5、泰勒公式展开，这是一个有用的公式，看图：

$$
\begin{aligned}
f(x) = {} & f(x_0) + f'(x_0)(x - x_0) \\
& + \frac{f''(x_0)}{2}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\
& + \frac{f^{(n+1)}\left(x_0 + \theta(x - x_0)\right)}{(n+1)!}(x - x_0)^{n+1} \quad (0 < \theta < 1).
\end{aligned}
$$

意义：可用n次多项式来近似表达函数$f(x)$，且误差是当$x \to x_0$时比$(x - x_0)^n$高阶的无穷小.

一、利用导数作近似计算

1. 近似计算

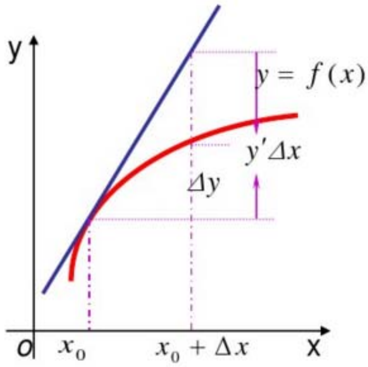
是用计算方法得到一定精度的计算结果.

若 $y=f(x)$ 在 x_0 可微，则有

$$\Delta y = f'(x_0)\Delta x + o(\Delta x),\ (\Delta x \to 0)$$

于是 $\Delta y \approx f'(x_0)\Delta x$,（$|\Delta x|$ 充分小）.

又 $\Delta y = f(x_0+\Delta x) - f(x_0)$,



故 $f(x_0+\Delta x) \approx f(x_0) + f'(x_0)\Delta x$,（$|\Delta x|$ 充分小）.