```
begin
    using StatsPlots    , StatsBase    , Symbolics    , Latexify    ,LinearAlgebra
end
```

最速下降法

神经网络设计(第二版):p132

$$x_1^2 + 25x_2^2$$

```
begin
    @variables ∇f, x₁,x₂,α, F(x₁,x₂)
    vars=[x₁,x₂]
    F=(x₁)^2+25*(x₂^2)
    x₀=[0.5,0.5]
    α=0.01
    ∇f=grad= Symbolics.gradient(F,[x₁,x₂])
    xk(x)=Pair.(vars,x)|>Dict
    g₀=substitute(∇f,xk(x₀))
    F

end

```

$[2x_1, 50x_2]$

```
∇f
```

H =

$$\begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}$$

```
H=Symbolics.hessian(F,[x₁,x₂])
```

steepest_descent (generic function with 1 method)

```
function steepest_descent(∇f, x, k_max;α=0.01)
     k=1
     while  k ≤ k_max
            Δ = substitute(∇f,xk(x))
            x -= α*Δ
            k += 1
     end
    return x
end
```

```
f1 (generic function with 1 method)
```

```
f1(k)=steepest_descent(∇f,x₀,k;α=0.01)
```

$[0.49,\ 0.25]$

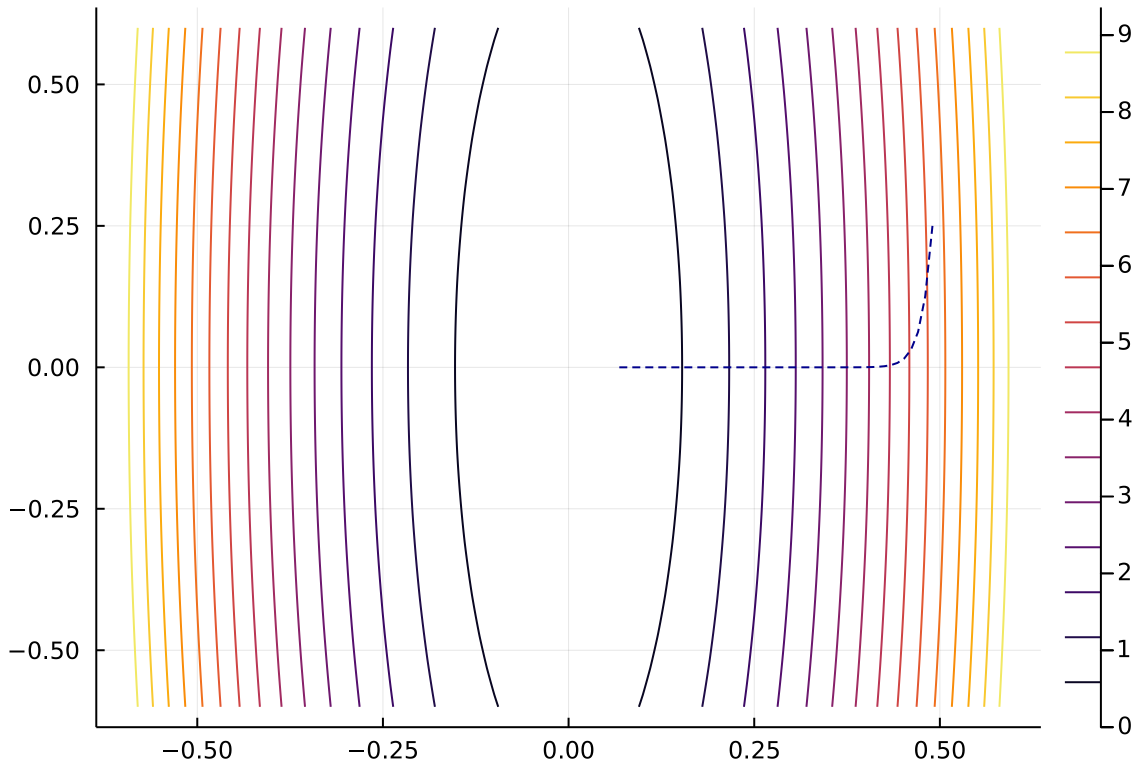
```
f1(1)
```

$[0.4802,\ 0.125]$

```
f1(2)
```

```
path1 =
 [[0.49, 0.25], [0.4802, 0.125], [0.470596, 0.0625], [0.461184, 0.03125], [0.45196, 0.0156
```

```
path1=[f1(k).|>d->Symbolics.value(d) for k in 1:100]  # 梯度的路径
```



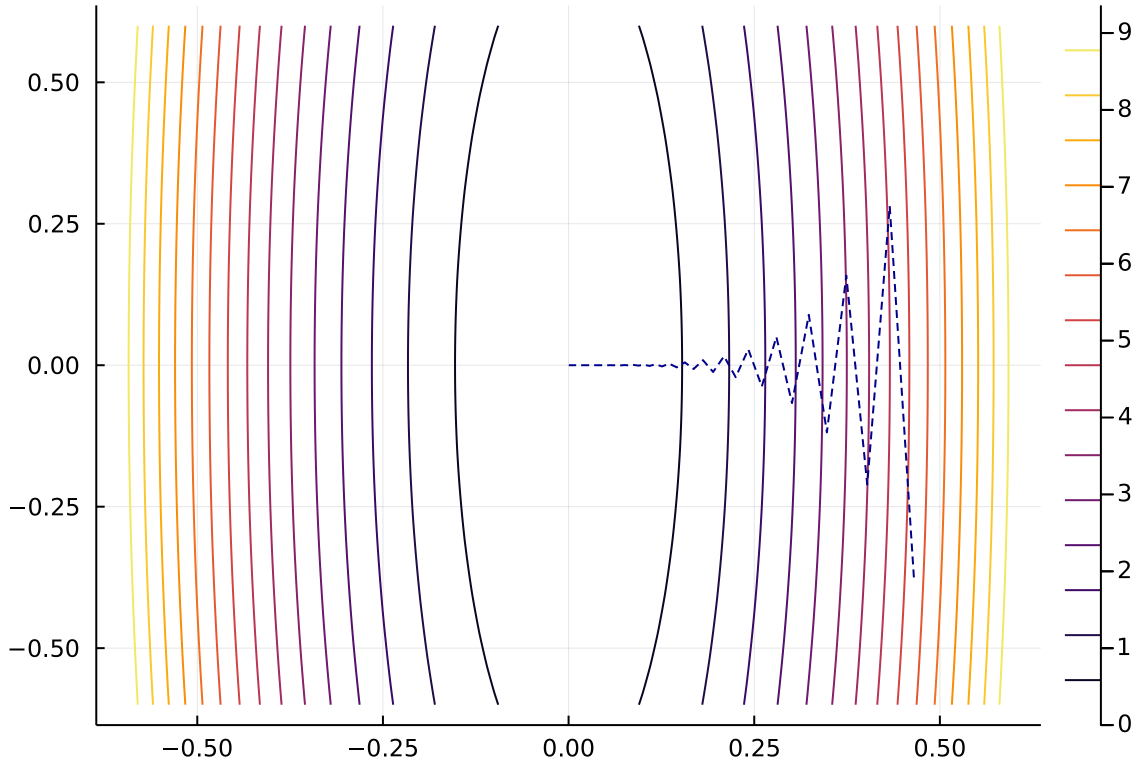

```
begin
    span=-0.6:0.01:0.6
    f(X)=X[1]^2+25*(X[2]^2)
    xs, ys=span, span
    zs=[f([x,y]) for x in xs, y in ys]
    contour(xs, ys,zs, label=false )
    xs1=[x[1] for x in path1]
    ys1=[x[2] for x in path1]
    plot!(xs1,ys1,label=false,ls=:dash,lc=:darkblue)
end
```

```
f2 (generic function with 1 method)
```

```
f2(k)=steepest_descent(∇f,x₀,k;α=0.035)  #当步长为 0.035 的高阶函数
```

```
path2 =
 [[0.465, -0.375], [0.43245, 0.28125], [0.402178, -0.210938], [0.374026, 0.158203], [0.347
```

```
path2=[f2(k).|>d->Symbolics.value(d) for k in 1:100]
```




```
let
    span=-0.6:0.01:0.6
    f(X)=X[1]^2+25*(X[2]^2)
    xs, ys=span, span
    zs=[f([x,y]) for x in xs, y in ys]
    contour(xs, ys,zs, label=false )
    xs2=[x[1] for x in path2]
    ys2=[x[2] for x in path2]
    plot!(xs2,ys2,label=false,ls=:dash,lc=:darkblue)
end
```

确定学习率α

学习率α,变大时的时候, 函数在向最小值收敛的时候是锯齿状的. 到底什么因素决定

$$x_{k+1} = x_k\,(I - A\alpha) - d\alpha$$

```
let
    @variables  x,F(x),A,d,c,xₖ,xₖ₊₁,α,I   #I为单位矩阵
    F=1//2*(x')*A*x+(d')*x+c   # 二次函数
    ∇f= Symbolics.gradient(F,[x])
    xₖ₊₁~xₖ-(α⋅∇f)
    xₖ₊₁~[I-α⋅A]⋅xₖ-α⋅d
end
```

这是一个动力系统, 可以参考线性代数及其应用的第五章内容. 一个动力系统要稳定, 矩阵特征值的绝对值需要小于1,利用黑塞矩阵就可以表示矩阵的特征值

$$|(1-\alpha\lambda_i)| < 1$$

对于有强极小点的二次函数,特征值为正, 可以得到不等式:

$$\alpha < \frac{2}{\lambda_i}$$

取最大特征值确定α:

$$\alpha < \frac{2}{\lambda_{max}}$$

```
alpha = 0.04
```

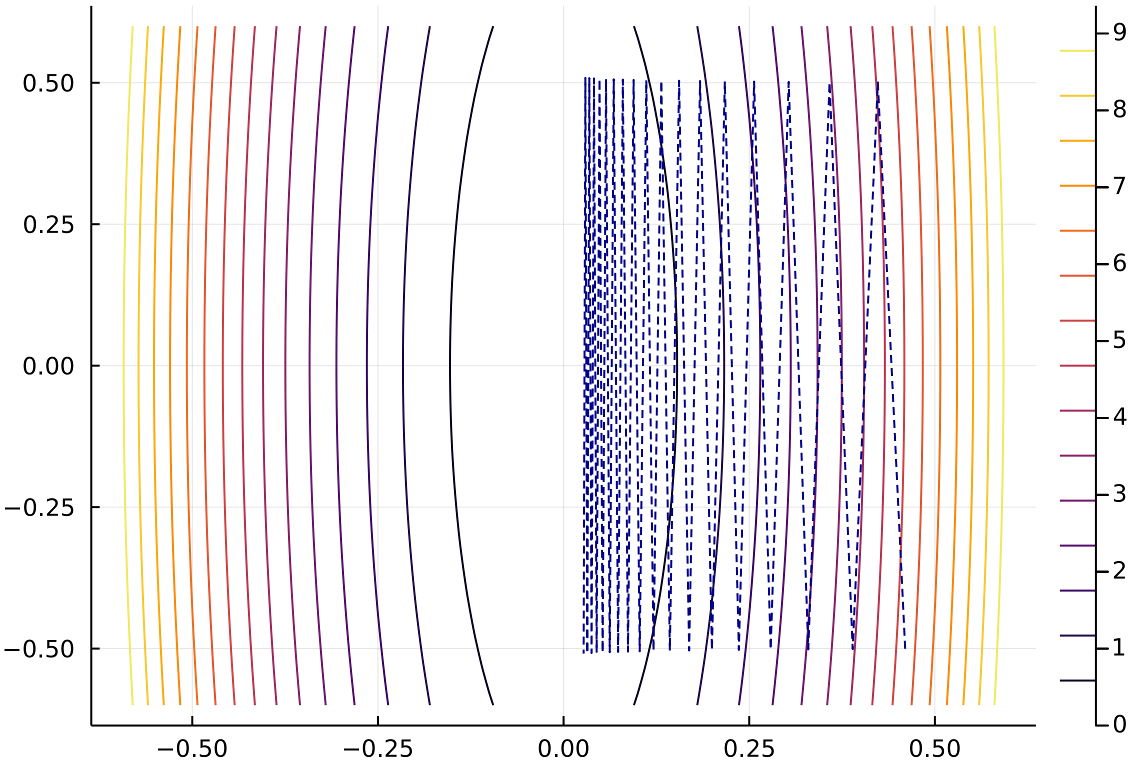
```
alpha= Symbolics.hessian(F,[x₁,x₂]).|>(d->Symbolics.value(d))|>eigen|>d->d.values|>d->max(d...)|>d->2/d
```

```
f3 (generic function with 1 method)
```

```
f3(k)=steepest_descent(∇f,x₀,k;α=0.04001)  # 当alpha超过 0.04时,不会收敛, 始终处于震荡状态
```

```
path3 =
[[0.45999, -0.50025], [0.423182, 0.5005], [0.389319, -0.50075], [0.358165, 0.501001], [0.
```

```
path3=[f3(k).|>d->Symbolics.value(d) for k in 1:35]
```




```
let
    span=-0.6:0.01:0.6
    f(X)=X[1]^2+25*(X[2]^2)
    xs, ys=span, span
    zs=[f([x,y]) for x in xs, y in ys]
    contour(xs, ys,zs, label=false )
    xs3=[x[1] for x in path3]
    ys3=[x[2] for x in path3]
    plot!(xs3,ys3,label=false,ls=:dash,lc=:darkblue)
end
```