```
begin
    using  Plots   , LaTeXStrings   ,Latexify   ,LinearAlgebra   ,Calculus
end
```

```
fun = powell (generic function with 1 method)
```

```
fun=powell(u)=(u[1]+10u[2])^2+5(u[3]-u[4])^2+(u[2]-2u[3])^4+10(u[1]-u[4])^4
```

```
∇f = #2 (generic function with 1 method)
```

```
∇f=Calculus.gradient(powell)  #函数梯度
```

```
H = #7 (generic function with 1 method)
```

```
H=hessian(powell)  # 函数黑塞矩阵
```

```
newtons_method (generic function with 2 methods)
```

```
function newtons_method(∇f, H, x, k_max, ε=0.005)
     k, Δ = 1, fill(Inf, length(x))

    while norm(Δ) > ε && k ≤ k_max
            Δ = H(x) \ ∇f(x)
            x -= Δ
            k += 1
    end
    return x
end
```

```
x₀ =  [3.0, -1.0, 0.0, 1.0]
```

```
x₀=[3,-1,0,1]|>float  # 值需要转位浮点数
```

```
[306.0, -144.0, -2.0, -310.0]
```

```
∇f(x₀)
```

```
4×4 Matrix{Float64}:
  482.0      19.9999         0.0      -480.0
   19.9999  212.0          -23.9999      0.000193776
    0.0     -23.9999        58.0         -9.99993
 -480.0       0.000193774   -9.99993   490.001
```

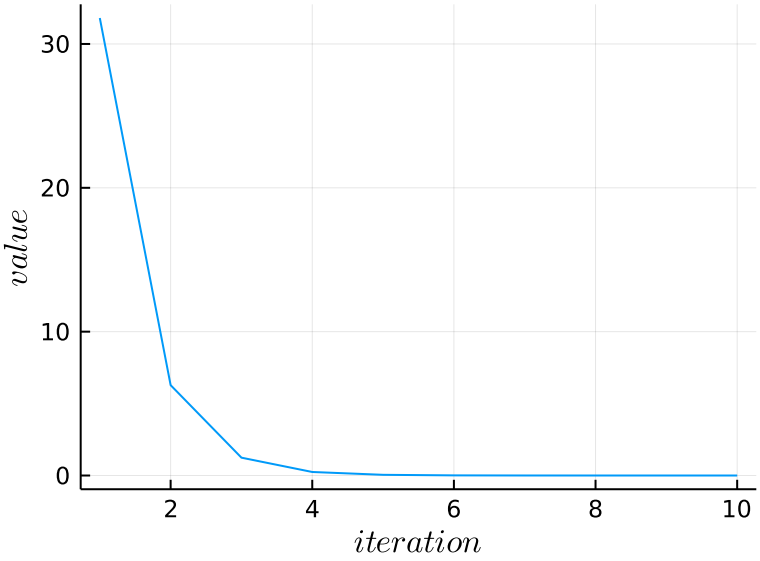
```
H(x₀)
```

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

```
iter(k)=newtons_method(∇f,H,x₀,k)|>fun
```

```
[31.8026, 6.28199, 1.24089, 0.245113, 0.0484175, 0.00956395, 0.00188917, 0.00037317, 7.37
```

```
begin
    span=1:10
    val=[iter(k) for k in  span]
end
```




```
plot(span,val,xlabel=L"iteration", ylabel=L"value",label=false,size=(400,300))
```