第一次上机作业

袁雨 PB20151804

一、实验目的

通过使用 C/C++ 语言实现两种函数求根的算法，并分析比较两种算法。

二、实验要求

使用“二分法”和“牛顿法”求解给定一元函数 f(x) 的根。要求如下：

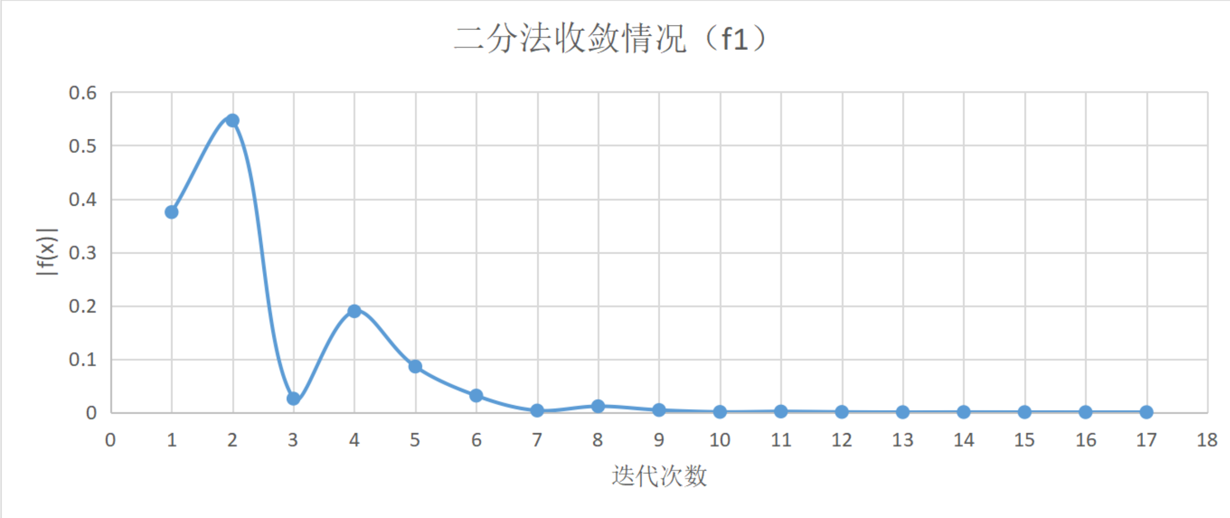
1. f(x) 从如下几个函数中，任选 2 个测试:
   - f(x) = (x − 1)^3 − x^2 + x，求根区间 [2, 3]
   - f(x) = (sin(x))^3 + (cos(x))^3，求根区间 [2, 3]
   - f(x) = e^x lnx − x2，求根区间 [1, 2]
   - f(x) = (x − 2)^5 − sin(x)，求根区间 [2, 3]
   - f(x) = cos(x) − e^x，求根区间 [−2, −1]
2. 终止条件统一设定为 |f(x)| < 10^-5。
3. 初始值在给定的求根区间内自行选择，应至少选择 2 个以作比较。

三、实验结果与分析

1. 二分法

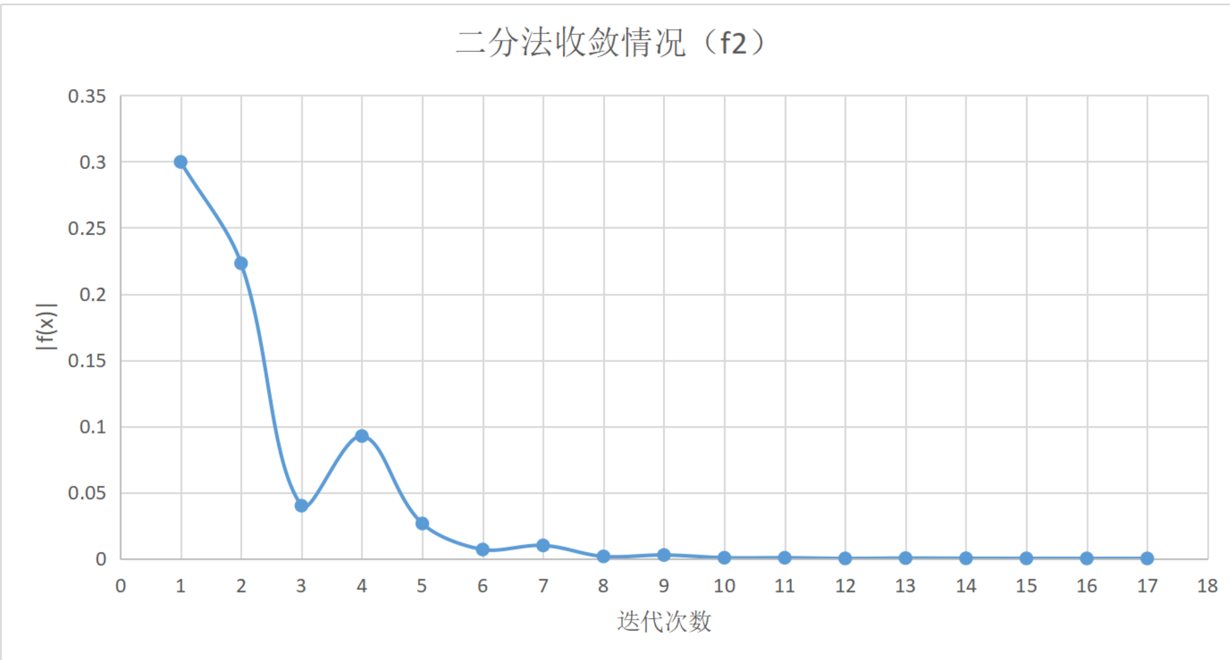
（1） f(x) = (x − 1)^3 − x^2 + x，求根区间 [2, 3]

```
input a=2
input b=3
iteration:1:f(2.500000000000)=-0.375000000000
iteration:2:f(2.750000000000)=0.546875000000
iteration:3:f(2.625000000000)=0.025390625000
iteration:4:f(2.562500000000)=-0.189208984375
iteration:5:f(2.593750000000)=-0.085601806641
iteration:6:f(2.609375000000)=-0.031040191650
iteration:7:f(2.617187500000)=-0.003059864044
iteration:8:f(2.621093750000)=0.011106431484
iteration:9:f(2.619140625000)=0.004008568823
iteration:10:f(2.618164062500)=0.000470676459
iteration:11:f(2.617675781250)=-0.001295512426
iteration:12:f(2.617919921875)=-0.000412647671
iteration:13:f(2.618041992188)=0.000028956956
iteration:14:f(2.617980957031)=-0.000191859726
iteration:15:f(2.618011474609)=-0.000081454971
iteration:16:f(2.618026733398)=-0.000026249905
iteration:17:f(2.618034362793)=0.000001353301
approximate root:2.618034362793

--------------------------------
```



（2） f(x) = (sin(x))^3 + (cos(x))^3，求根区间 [2, 3]

```
input a=2
input b=3
iteration:1:f(2.500000000000)=-0.299844771624
iteration:2:f(2.250000000000)=0.223165243864
iteration:3:f(2.375000000000)=-0.039880756289
iteration:4:f(2.312500000000)=0.092542655766
iteration:5:f(2.343750000000)=0.026395343244
iteration:6:f(2.359375000000)=-0.006746823434
iteration:7:f(2.351562500000)=0.009825759567
iteration:8:f(2.355468750000)=0.001539526740
iteration:9:f(2.357421875000)=-0.002603673143
iteration:10:f(2.356445312500)=-0.000532074424
iteration:11:f(2.355957031250)=0.000503726478
iteration:12:f(2.356201171875)=-0.000014173989
iteration:13:f(2.356079101563)=0.000244776253
iteration:14:f(2.356140136719)=0.000115301133
iteration:15:f(2.356170654297)=0.000050563569
iteration:16:f(2.356185913086)=0.000018194791
iteration:17:f(2.356193542480)=0.000002010401
approximate root:2.356193542480

--------------------------------
```

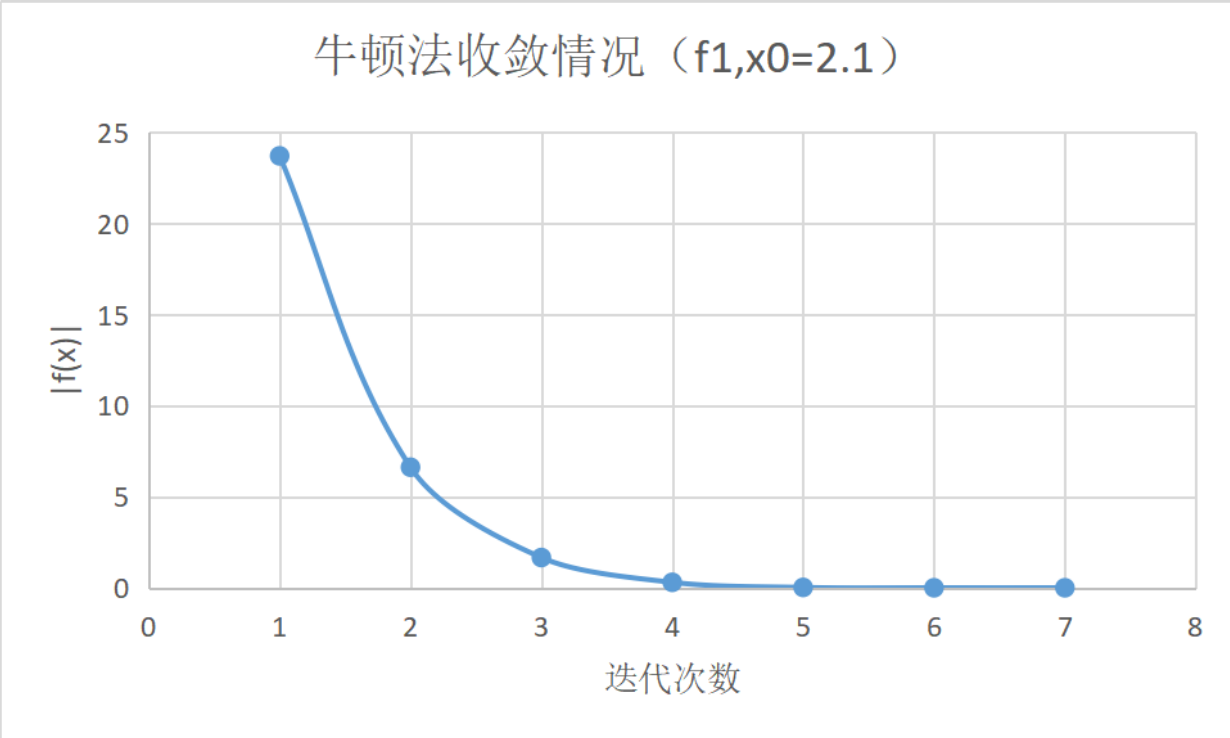


2. 牛顿法

（1）f(x) = (x − 1)^3 − x^2 + x，求根区间 [2, 3]

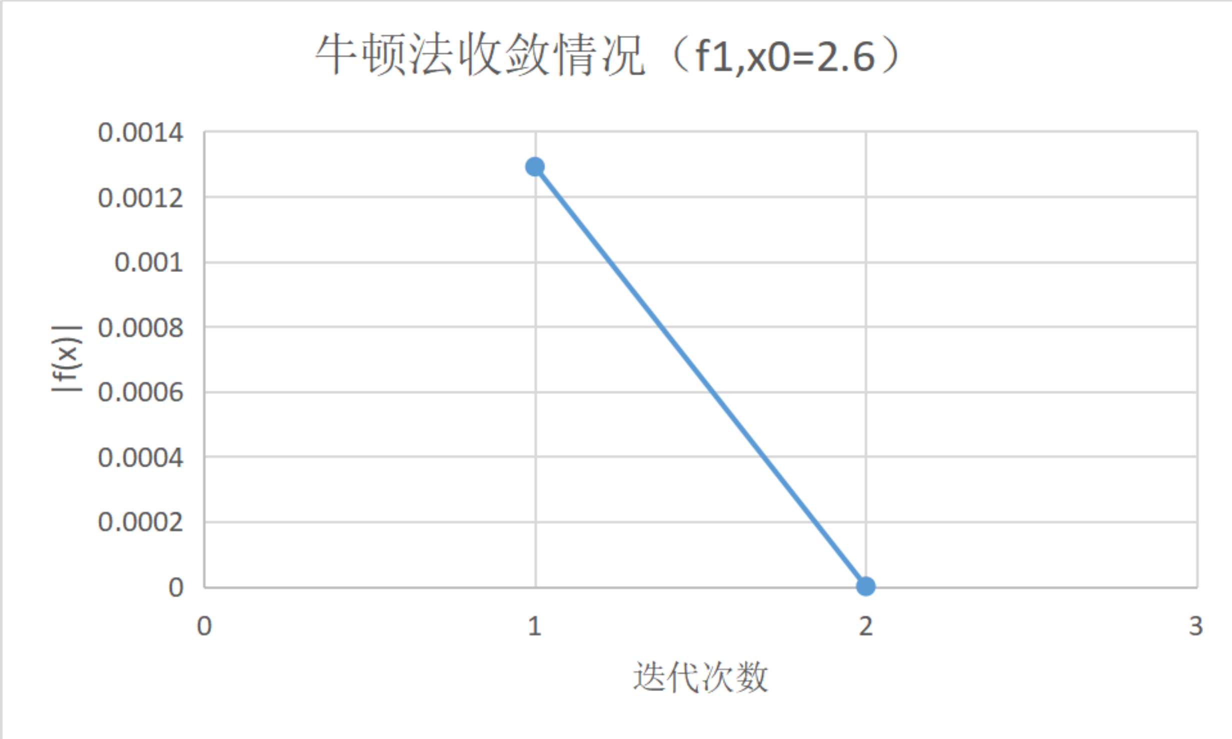
①取x0=2.1:

```
input x0=2.1
iteration:1:f(4.376746177673)=23.723899841309
iteration:2:f(3.479939460754)=6.621835708618
iteration:3:f(2.949786186218)=1.660983800888
iteration:4:f(2.694463253021)=0.299483597279
iteration:5:f(2.623574256897)=0.020163347945
iteration:6:f(2.618066310883)=0.000116946598
iteration:7:f(2.618033885956)=-0.000000371912
approximate root:2.618033885956

--------------------------------
```



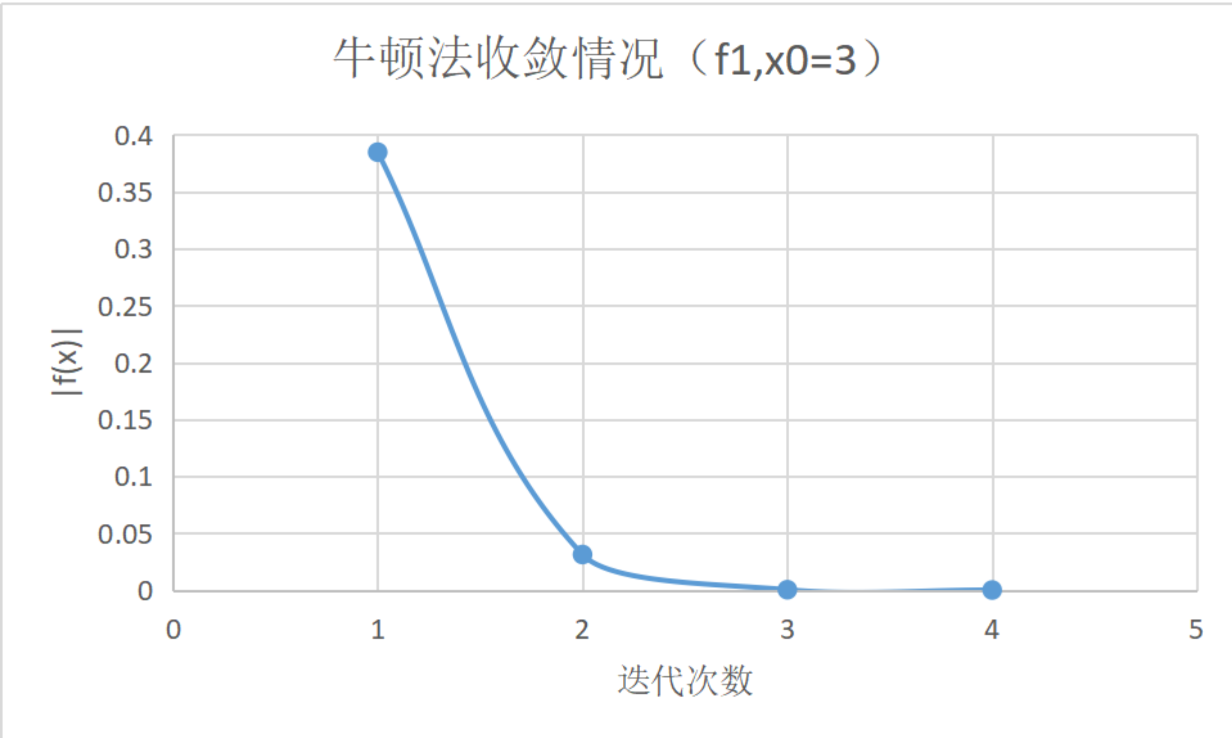
②取x0=2.6:

```
input x0=2.6
iteration:1:f(2.618390798569)=0.001291440800
iteration:2:f(2.618034124374)=0.000000490694
approximate root:2.618033885956

--------------------------------
```



③取x0=3:

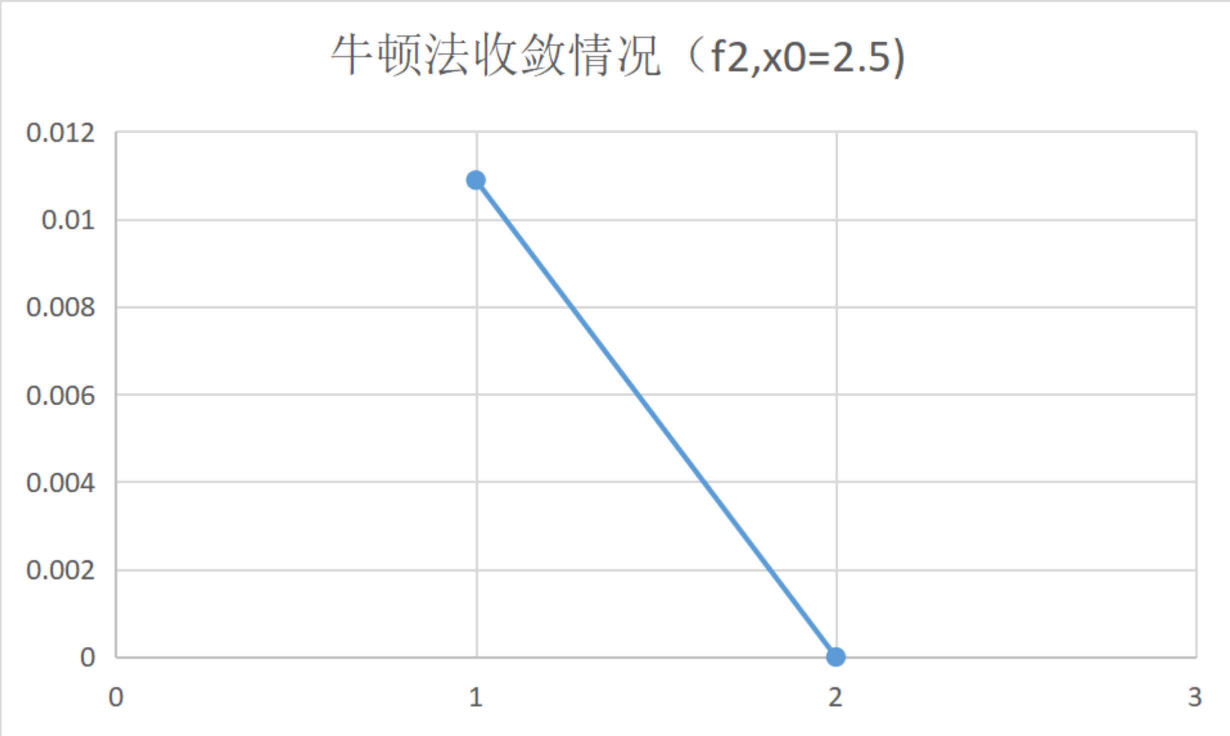
```
input x0=3
iteration:1:f(2.714285612106)=0.384839206934
iteration:2:f(2.626578092575)=0.031194837764
iteration:3:f(2.618110656738)=0.000277410029
iteration:4:f(2.618033885956)=-0.000000371912
approximate root:2.618033885956
```



（2） f(x) = (sin(x))^3 + (cos(x))^3，求根区间 [2, 3]

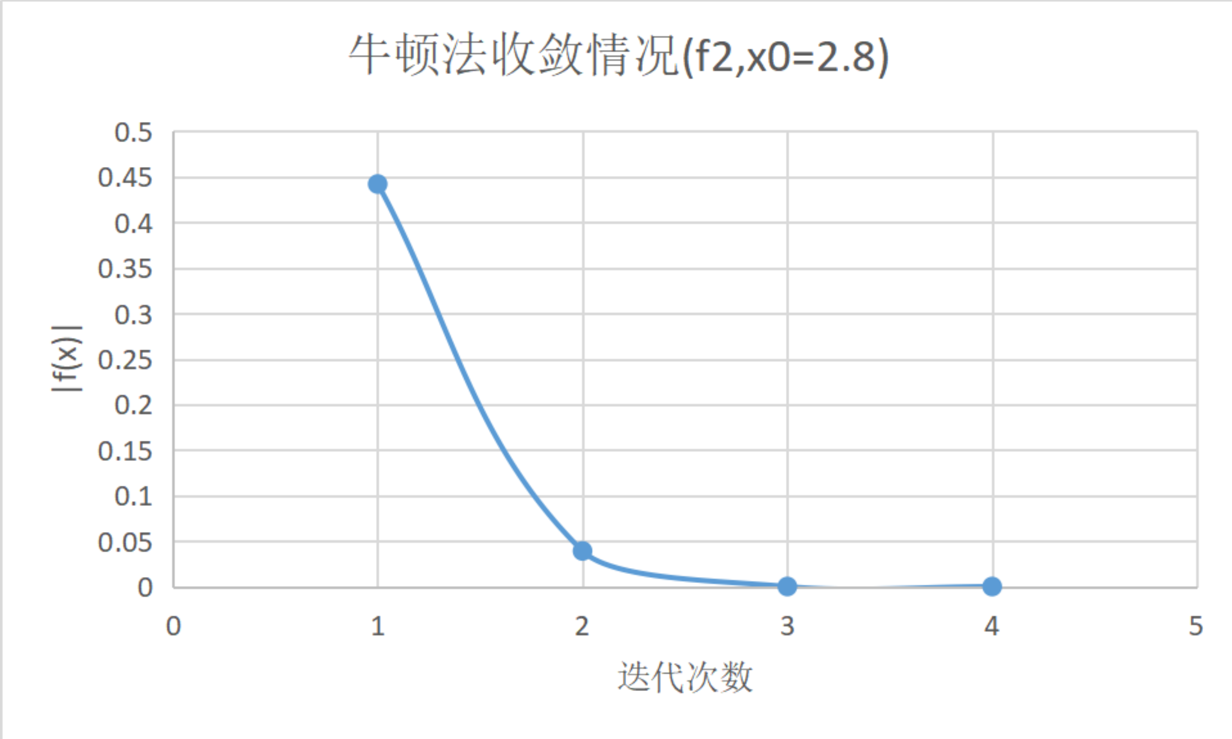
①取x0=2.5:

```
input x0=2.5
iteration:1:f(2.351059675217)=0.010892348364
iteration:2:f(2.356194734573)=-0.000000518410
approximate root:2.356194496155

--------------------------------
```



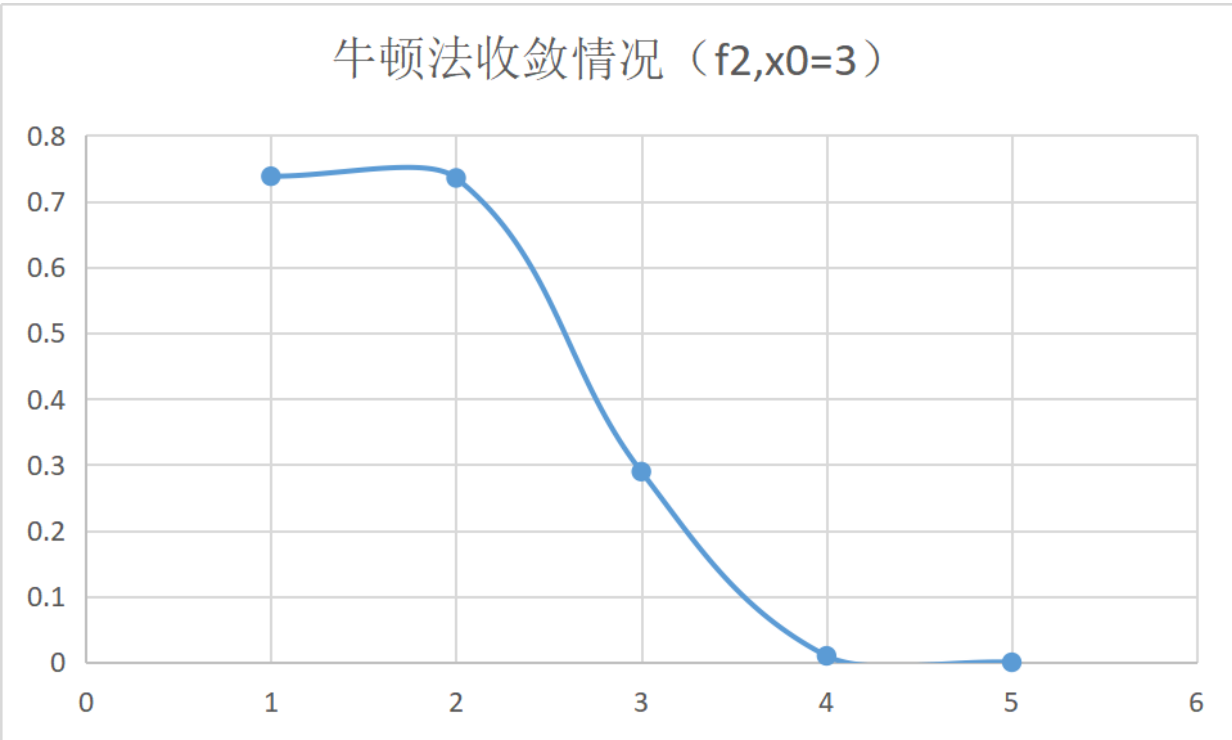
②取x0=2.8:

```
input x0=2.8
iteration:1:f(2.139421463013)=0.442181557417
iteration:2:f(2.374623537064)=-0.039082847536
iteration:3:f(2.356184005737)=0.000022240887
iteration:4:f(2.356194496155)=-0.000000012648
approximate root:2.356194496155

--------------------------------
```



③取x0=3:

```
input x0=3
iteration:1:f(0.959256649017)=0.738127052784
iteration:2:f(-1.180283546448)=-0.735559642315
iteration:3:f(-0.646686851978)=0.289570152760
iteration:4:f(-0.789994835854)=-0.009750843048
iteration:5:f(-0.785398006439)=0.000000332959
approximate root:-0.785398185253

----------------------------------
```



对比所取三个初值的迭代情况，可以发现，取的初值越接近方程的根，迭代次数越少。
对比牛顿法与二分法，可以发现，牛顿法比二分法收敛更快。