# 第二次上机作业

**问题叙述：**

- 编写非线性方程求根的不动点迭代算法程序
- 从不同的初始点开始，分别采用以下三种迭代方式求解方程 $x^2 - x - 1 = 0$ 的正根，记录迭代过程并说明原因。

$$x = x^2 - 1 \tag{1}$$

$$x = 1 + \frac{1}{x} \tag{2}$$

$$x = \sqrt{x+1} \tag{3}$$

**Matlab 代码：**

```
function [k,p,ea,P,et,x] = fixpt(g,p0,tol,max1)
P(1)=p0;
real=1.618033988749895;
et(1)=abs((P(1)-real)/real);
for k=2:max1
    P(k)=feval(g,P(k-1));%不动点迭代
    err=abs(P(k)-P(k-1));
    ea(k)=err/(abs(P(k))+eps);%计算相对误差
    p=P(k);
    et(k)=abs((P(k)-real)/real);
    x(k)=k;
    if ea(k)<tol,break;end
end
if k == max1
    disp('maximum number of iterations exceeded')
end
P=P';%记录每次迭代的值
ea=ea';
et=et';
end
```

```
clc,clear,close all
start=2.5;%start point
tol=eps;
```

```
4
5  %%algorithm1
6  [k1,p1,ea1,P1,et1,x]=fixpt(@f1,start,tol,10000);
7  semilogy(x,P1);
8  xlabel('迭代次数');
9  ylabel('迭代值');
10
11 %%algorithm2
12 [k2,p2,ea2,P2,et2,x]=fixpt(@f2,start,tol,10000);
13 semilogy(x,P2);
14 xlabel('迭代次数');
15 ylabel('迭代值');
16
17 %%algorithm3
18 [k3,p3,ea3,P3,et3,x]=fixpt(@f3,start,tol,10000);
19 semilogy(x,P3);
20 xlabel('迭代次数');
21 ylabel('迭代值');
22
23 function f=f1(x)
24 f=x^2-1;
25 end
26
27 function f=f2(x)
28 f=1+1/x;
29 end
30
31 function f=f3(x)
32 f=sqrt(x+1);
33 end
```

**程序运行结果：**

第一种算法：

无法收敛，输出“maximum number of iterations exceeded”。

表 1: 起始点 x=1.5

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | 1.500000000000000 | Inf | 0.072949016875158 |
| 2 | 1.250000000000000 | 0.200000000000000 | 0.227457514062631 |
| 3 | 0.562500000000000 | 1.222222222222222 | 0.652355881328184 |
| 4 | -0.683593750000000 | 1.822857142857142 | 1.422484171996999 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 22 | -0.999999999999989 | 0.999999894563900 | 1.618033988749888 |
| 23 | -2.220446049250313e-14 | 4.459009532049848e+13 | 1.000000000000014 |
| 24 | -1 | 0.999999999999978 | 1.618033988749895 |
| 25 | 0 | 4.503599627370496e+15 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ |

表 2: 起始点 x=-2

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | -2 | Inf | 2.236067977499790 |
| 2 | 3 | 1.666666666666667 | 0.854101966249685 |
| 3 | 8 | 0.625000000000000 | 3.944271909999159 |
| 4 | 63 | 0.873015873015873 | 37.936141291243370 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 10 | 1.426569025399668e+115 | 1 | 8.816681449948065e+114 |
| 11 | 2.035099184229757e+230 | 1 | 1.257760466331174e+230 |
| 12 | inf | NaN | inf |
| 13 | inf | NaN | inf |
| ⋮ | ⋮ | ⋮ | ⋮ |

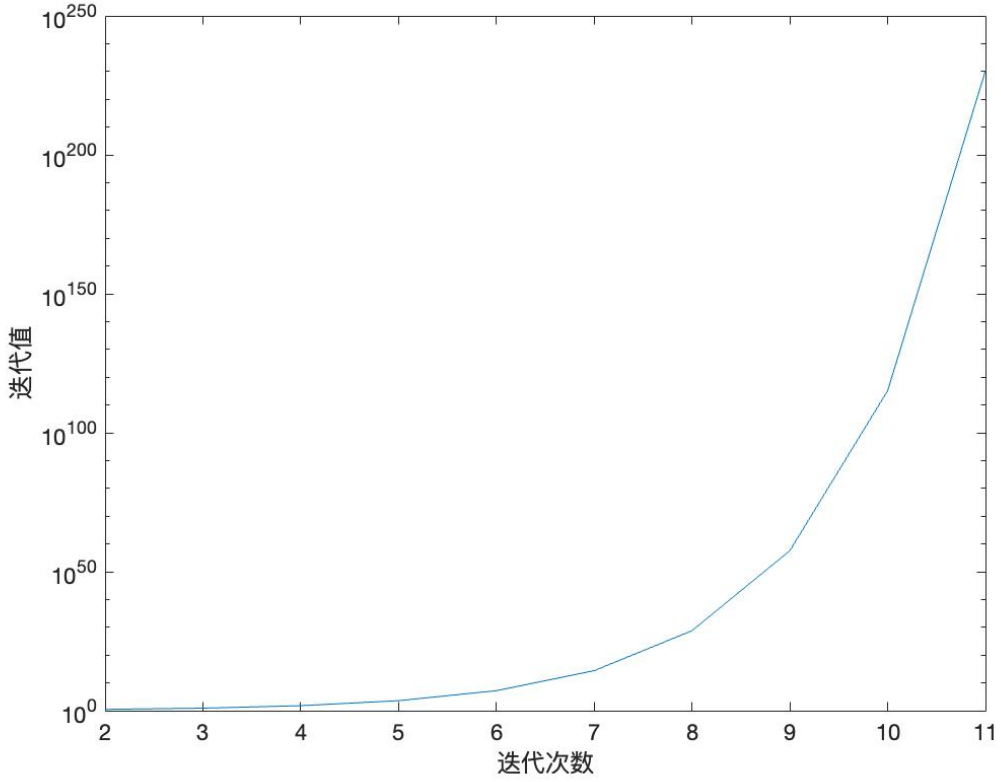


图 1: 起始点 x=-2

表 3: 起始点 x=2.5

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | 2.500000000000000 | Inf | 0.545084971874737 |
| 2 | 5.250000000000000 | 0.523809523809524 | 2.244678440936948 |
| 3 | 26.562500000000000 | 0.802352941176471 | 15.416527826169082 |
| 4 | 7.045664062500000e+02 | 0.962299508230350 | 4.344459863938664e+02 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 9 | 1.359840060999618e+91 | 1 | 8.404273769614945e+90 |
| 10 | 1.849164991499446e+182 | 1 | 1.142846815553068e+182 |
| 11 | inf | NaN | inf |
| 12 | inf | NaN | inf |
| ⋮ | ⋮ | ⋮ | ⋮ |

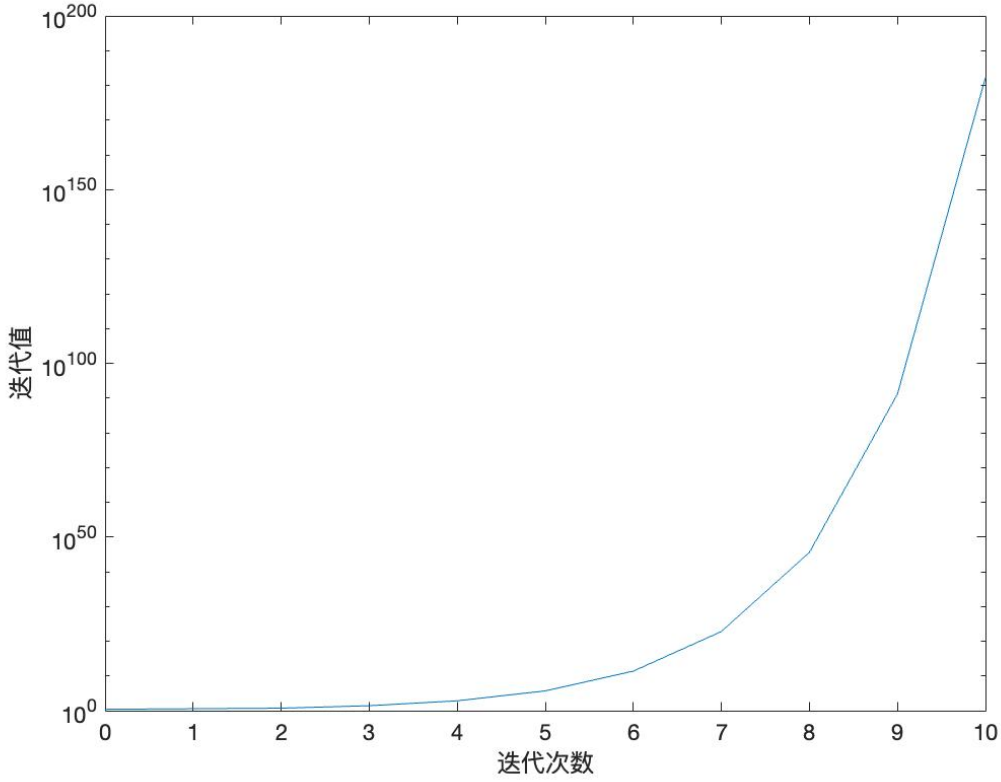


图 2: 起始点 x=2.5

第二种算法:

表 4: 起始点 x=1.5

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | 1.500000000000000 | Inf | 0.072949016875158 |
| 2 | 1.666666666666667 | 0.100000000000000 | 0.030056647916491 |
| 3 | 1.600000000000000 | 0.041666666666667 | 0.011145618000168 |
| 4 | 1.625000000000000 | 0.015384615384615 | 0.004305231718579 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 36 | 1.618033988749895 | 6.861555643110580e-16 | 1.372311128622116e-16 |
| 37 | 1.618033988749895 | 1.372311128622116e-16 | 0 |

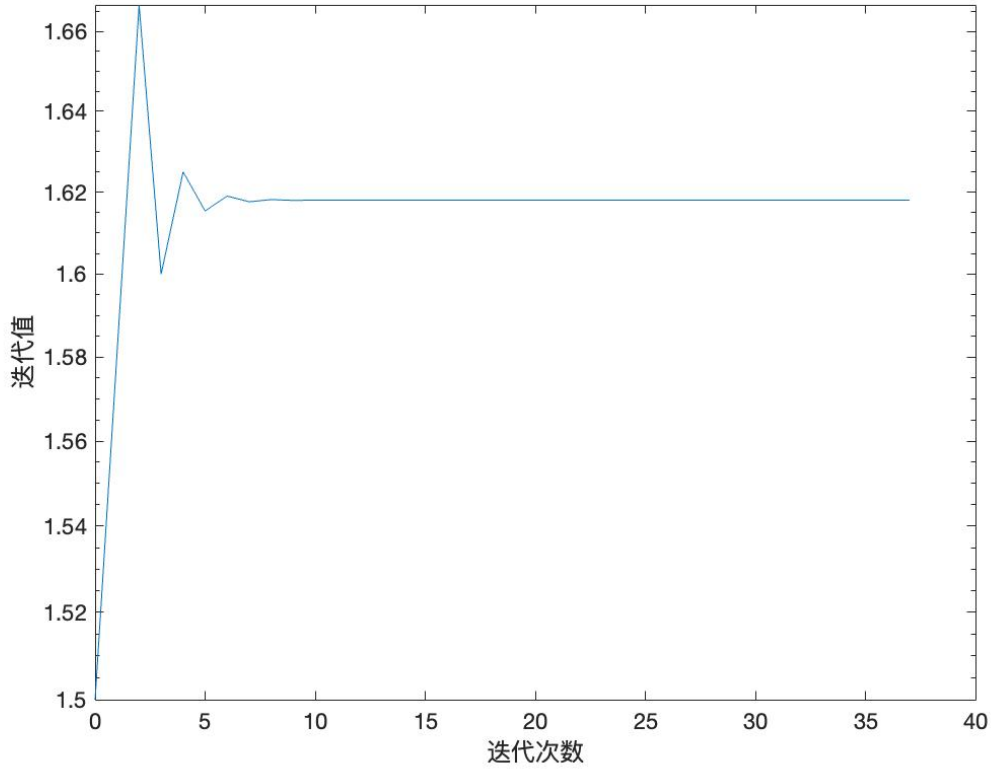


图 3: 起始点 x=1.5

表 5: 起始点 x=-2

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | -2 | Inf | 2.236067977499790 |
| 2 | 0.500000000000000 | 4.999999999999998 | 0.690983005625053 |
| 3 | 3 | 0.833333333333333 | 0.854101966249685 |
| 4 | 1.333333333333333 | 1.250000000000000 | 0.175954681666807 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 41 | 1.618033988749895 | 2.744622257244233e-16 | 0 |
| 42 | 1.618033988749895 | 0 | 0 |

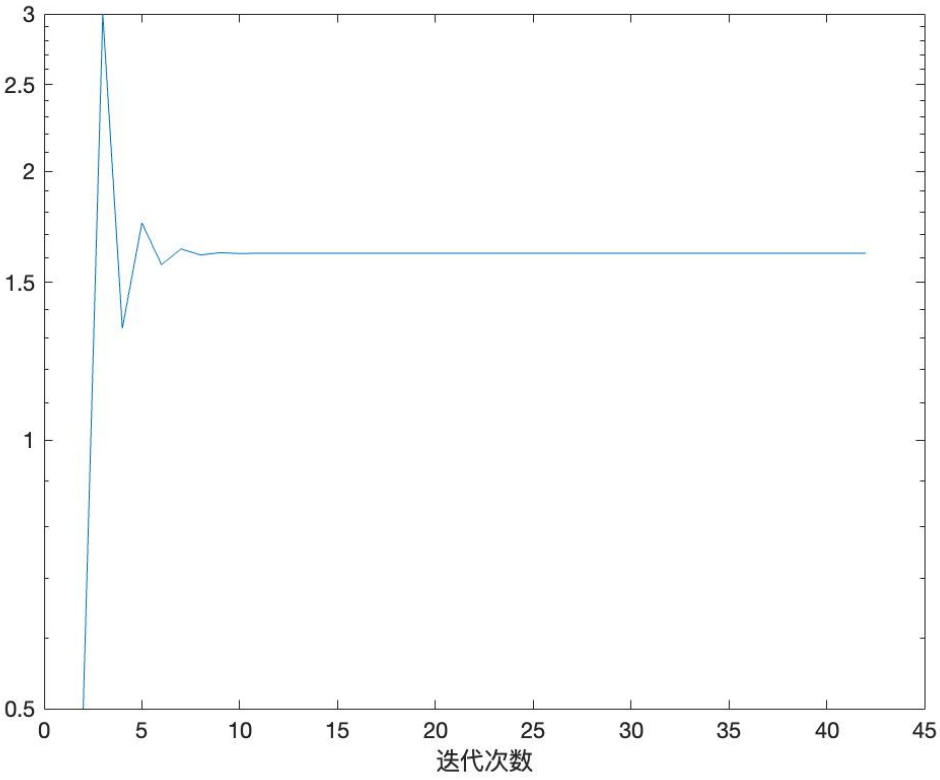


图 4: 起始点 x=-2

表 6: 起始点 x=2.5

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | 2.500000000000000 | Inf | 0.545084971874737 |
| 2 | 1.400000000000000 | 0.785714285714286 | 0.134752415750147 |
| 3 | 1.714285714285714 | 0.183333333333333 | 0.059486837856963 |
| 4 | 1.583333333333333 | 0.082706766917293 | 0.021446184479333 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 38 | 1.618033988749895 | 4.116933385866349e-16 | 1.372311128622116e-16 |
| 39 | 1.618033988749895 | 1.372311128622116e-16 | 0 |

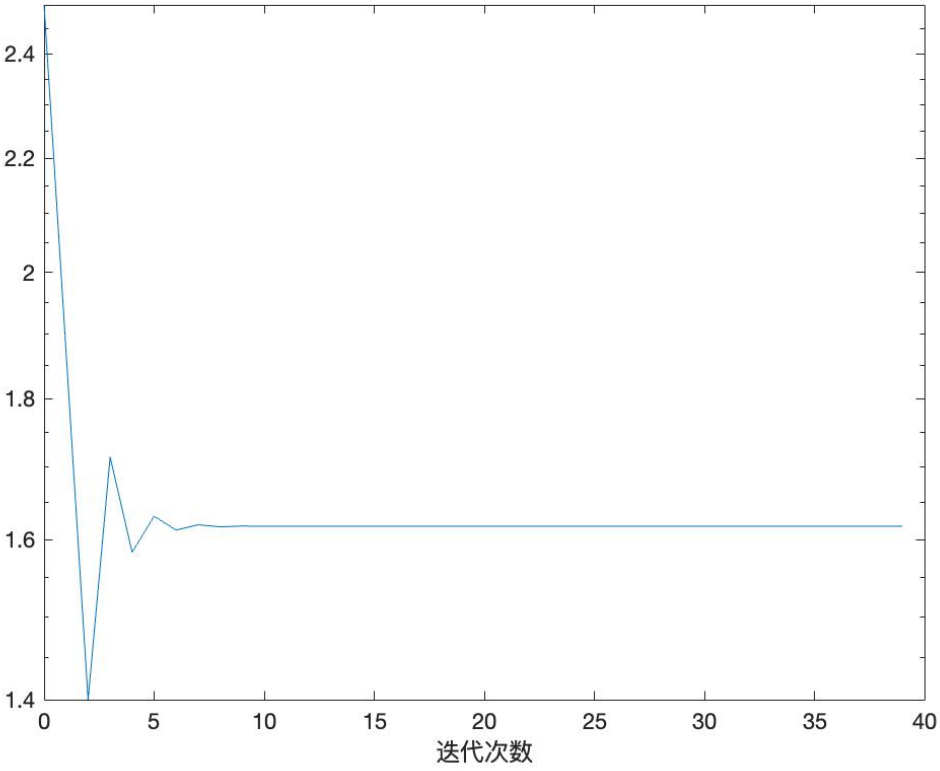


图 5: 起始点 x=2.5

第三种算法:

表 7: 起始点 x=1.5

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | 1.500000000000000 | Inf | 0.072949016875158 |
| 2 | 1.581138830084190 | 0.051316701949486 | 0.022802462075726 |
| 3 | 1.606592303630324 | 0.015843144205669 | 0.007071350292469 |
| 4 | 1.614494442118128 | 0.004894497176117 | 0.002187560123197 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 30 | 1.618033988749895 | 2.744622257244233e-16 | 1.372311128622116e-16 |
| 31 | 1.618033988749895 | 1.372311128622116e-16 | 0 |

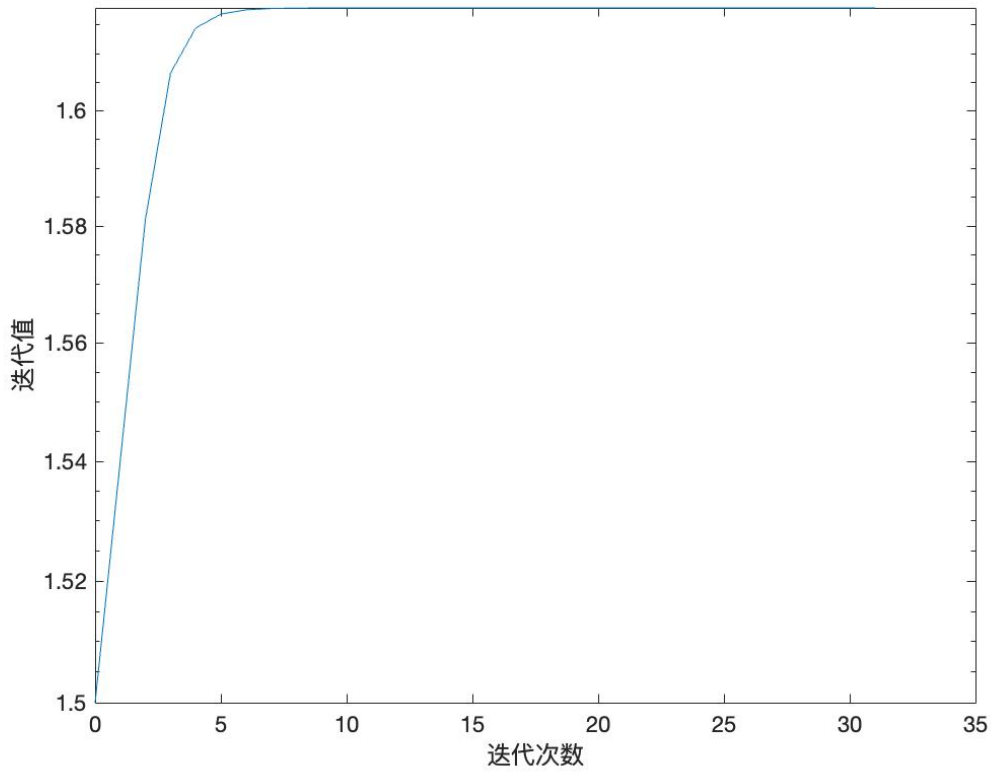


图 6: 起始点 x=1.5

表 8: 起始点 x=-0.8

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | -0.800000000000000 | Inf | 1.494427190999916 |
| 2 | 0.447213595499958 | 2.788854381999831 | 0.723606797749979 |
| 3 | 1.203001910015091 | 0.628251965539815 | 0.256503931079631 |
| 4 | 1.484251296113664 | 0.189489062152070 | 0.082682251155671 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 33 | 1.618033988749895 | 4.116933385866349e-16 | 1.372311128622116e-16 |
| 34 | 1.618033988749895 | 1.372311128622116e-16 | 0 |

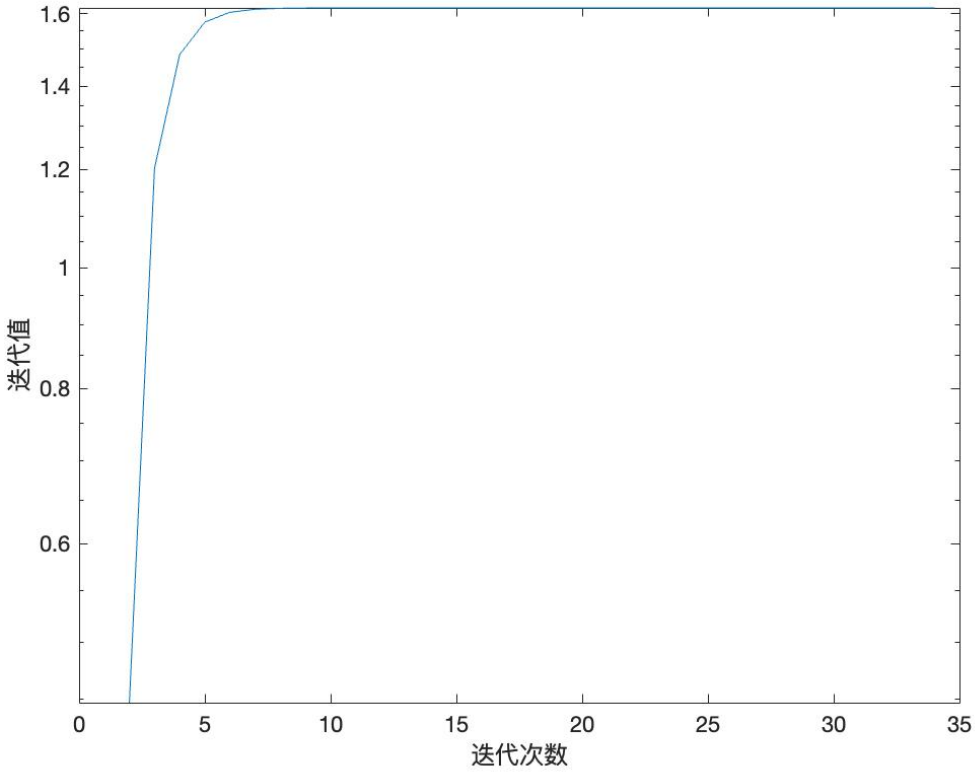


图 7: 起始点 x=-0.8

表 9: 起始点 x=2.5

| 序号 | 当前计算值 $x$ | 近似百分比相对误差 $\varepsilon_a$ | 真百分比相对误差 $\varepsilon_s$ |
|---|---|---|---|
| 1 | 2.500000000000000 | Inf | 0.545084971874737 |
| 2 | 1.870828693386971 | 0.336306209562122 | 0.156235719641703 |
| 3 | 1.694351998076837 | 0.104155863427695 | 0.047167123717782 |
| 4 | 1.641448140538359 | 0.032229990233579 | 0.014470741623020 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 31 | 1.618033988749895 | 5.489244514488464e-16 | 1.372311128622116e-16 |
| 32 | 1.618033988749895 | 1.372311128622116e-16 | 0 |

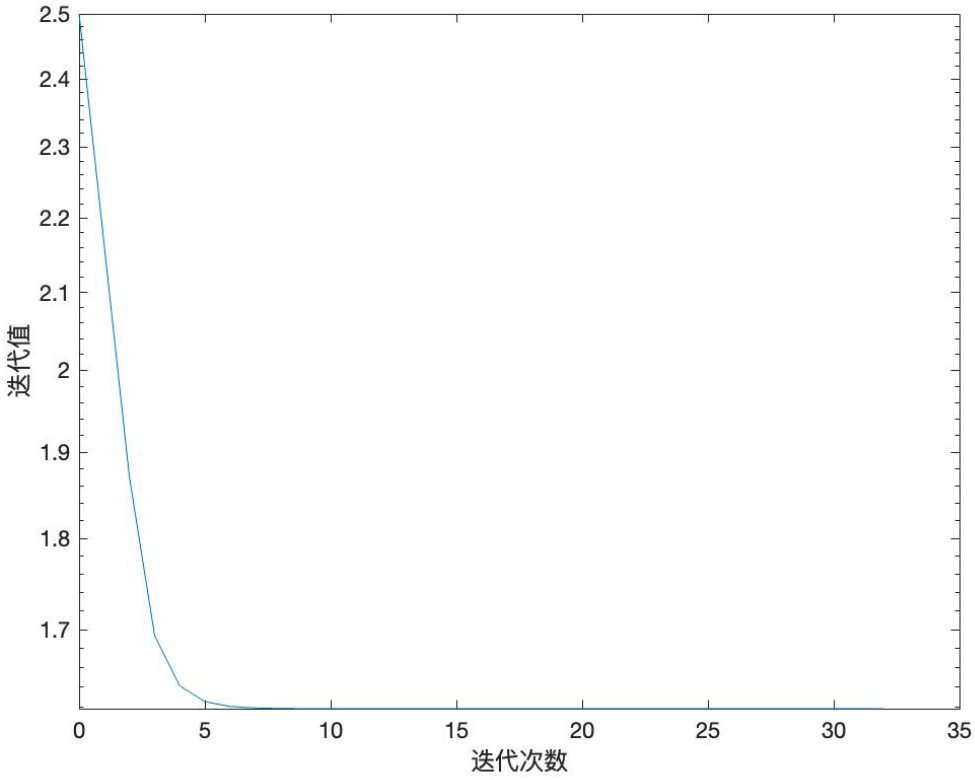


图 8: 起始点 x=2.5

**心得体会：**

对于第一问，定义当前迭代结果的近似相对误差：

$$\varepsilon_a = \frac{\text{当前近似值} - \text{前一近似值}}{\text{当前近似值}} \times 100\% \tag{4}$$

当 $\varepsilon_a < eps$(机器精度) 或迭代次数达到上限时停止迭代。

对于第二问，取迭代到最后一次的值作为真值计算误差。可以发现，对于第一种算法，当初值的绝对值小于真值 $x_0$ 时，迭代到最后会进入 0 和-1 之间循环；当初值的绝对值大于真值 $x_0$ 时，迭代到最后会趋近于正无穷。无论哪种情况，该算法都无法收敛。这是因为在零点附近 $g(x) = x^2 - 1$ 的导数大于 1，出现了发散的情况。同时又因为 $g(x)$ 有两个二阶不动点 0 和-1，因此若初值绝对值小于 $x_0$, 最后会稳定在这两个点之间循环。

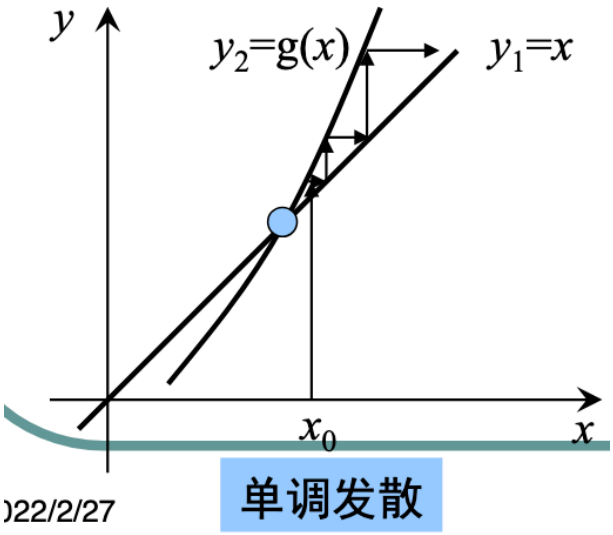


图 9: 单调发散

对于第二种算法，由图像可以发现迭代值在真值附近上下跳动最后收敛于真值。这是因为在零点附近 $g(x) = 1 + \frac{1}{x}$ 的导数小于 0 且大于-1，迭代值振荡收敛于真值。

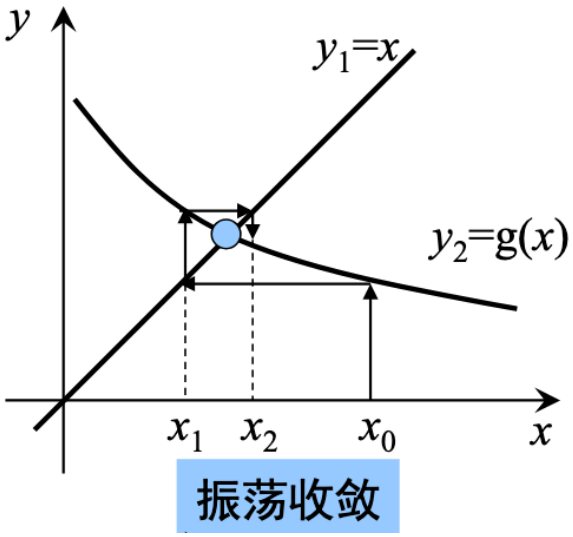


图 10: 振荡收敛

对于第三种算法，由图像可以发现迭代值逐步趋近于真值且一直在真值得同一边。这是因为在零点附近 $g(x) = \sqrt(x+1)$ 的导数大于 0 且小于 1，迭代值单调收敛于真值。

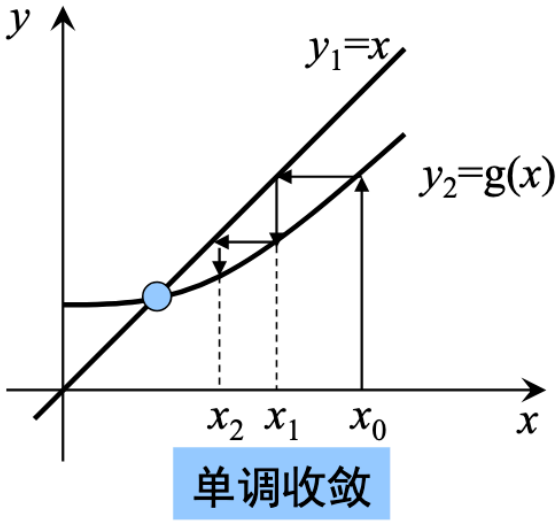


图 11: 单调收敛