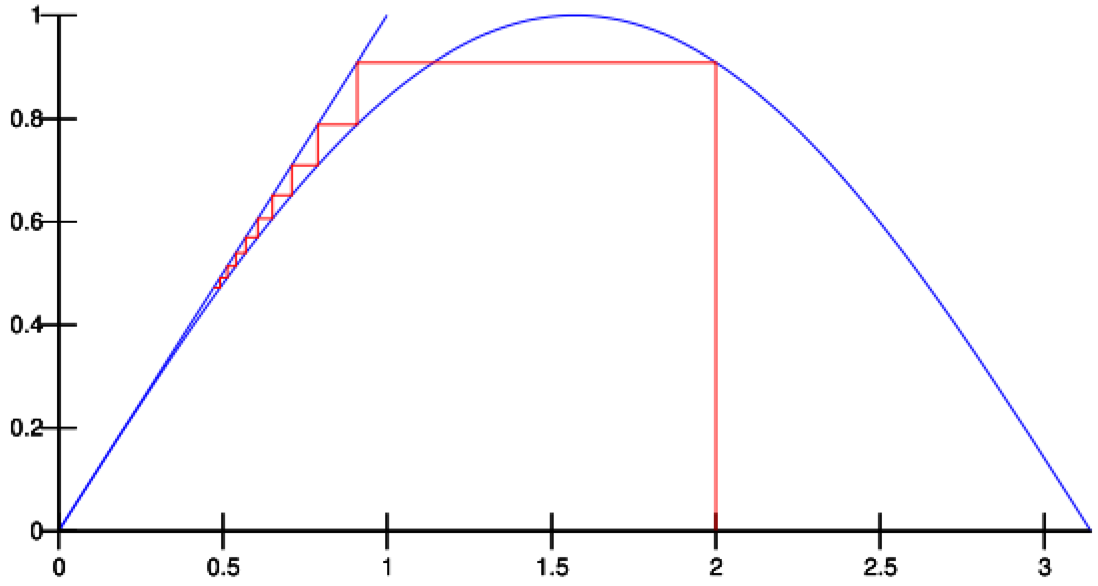
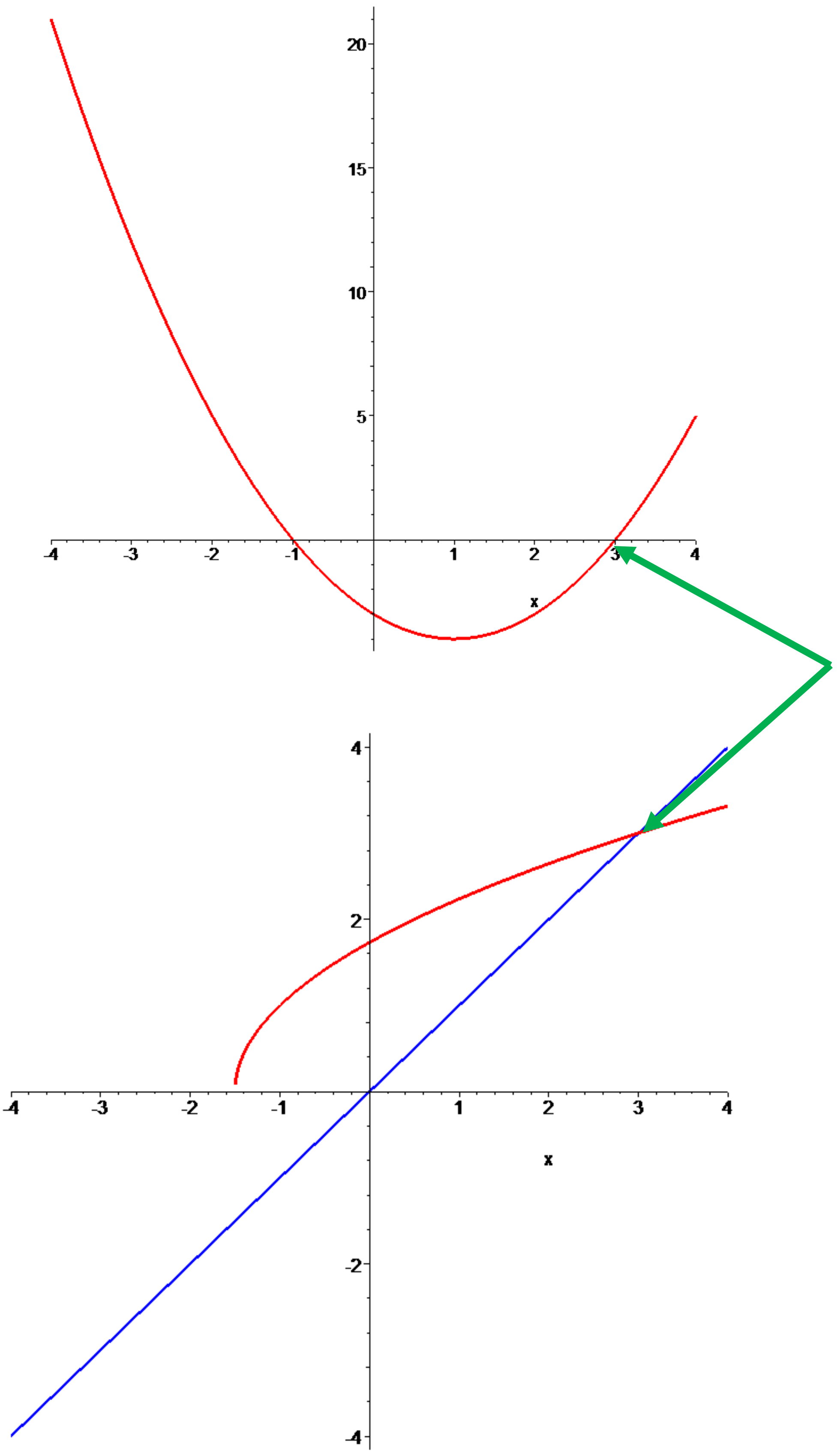
Fixed Point Iteration:

$x = g(x)$ Method


1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3



Compare figures

The method known as fixed point iteration (also call $x = g(x)$ method) can be a useful way to get a root $f(x)=0$. This method is also basis for some important theory to use the method, we rearrange $f(x)$ into an equivalent form $x = g(x)$, which is usually can be done in several ways.

Observe that if $f(r)=0$, where r is a root of $f(x)$, it follows that

$r = g(r)$ **, r is said to be a fixed for the function g.**

The iterative form

$$x_{n+1} = g(x_n) \qquad n = 0,1,2,3,\ldots$$

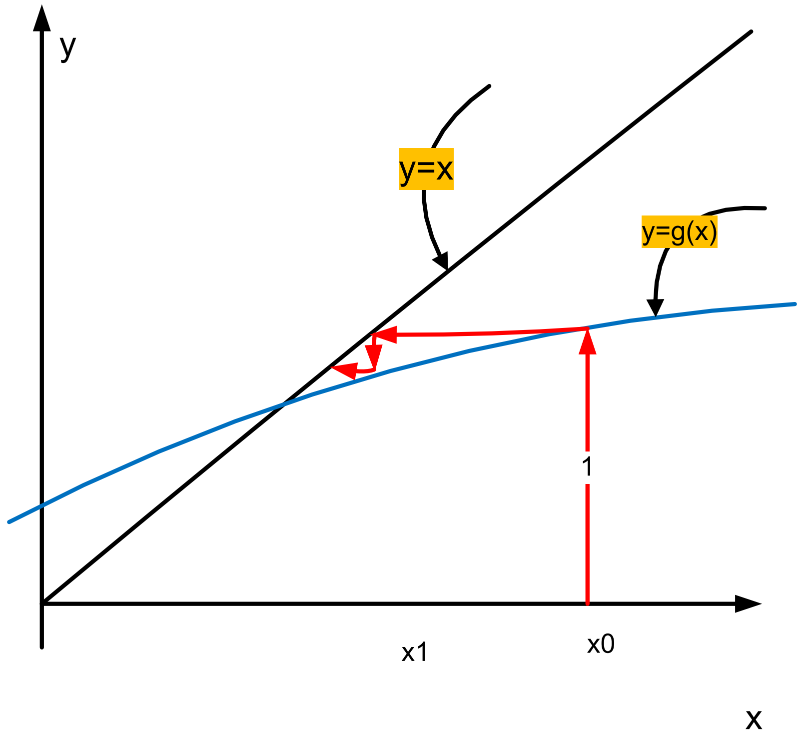
Definition: ***A fixed point of a function $g(x)$ is a real number P such that $P = g(P)$. G***

Geometrically, the fixed points of a function $y = g(x)$ are the points of intersection $y = g(x)$ and $y = x$.

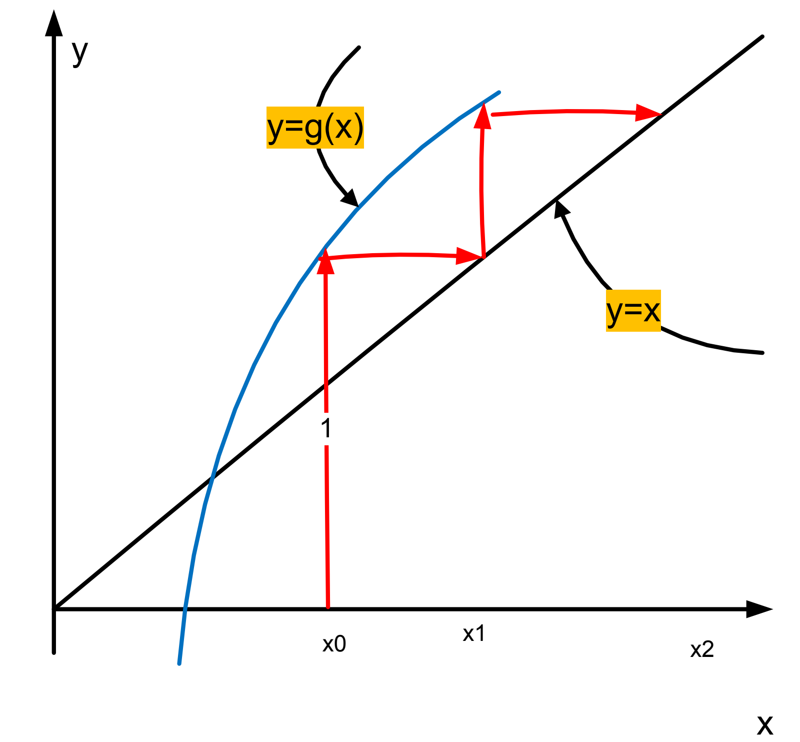
Theorem: ***Assume that g is a continuous function and that $\{p_n\}_{n=0}^{\infty}$ is a sequence generated by fixed point iteration.***

If $\lim_{n\to\infty} p_n = P$

then $\lim_{n\to\infty} p_{n+1} = P$, ***then P is a fixed point of $g(x)$.***



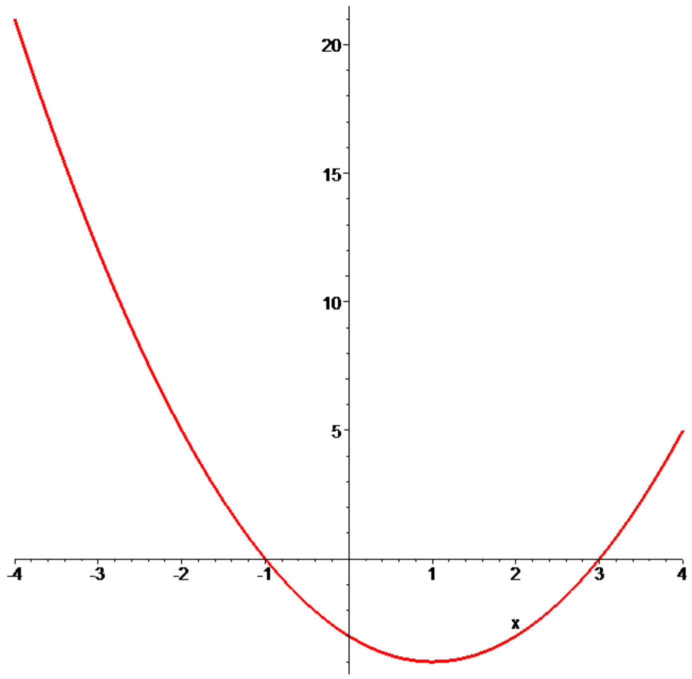
Convergent



Divergent

Example:

$$f(x) = x^2 - 2x - 3 = 0$$



$f(x)$ **is easy to factor to show roots at x=-1 and x=3.**

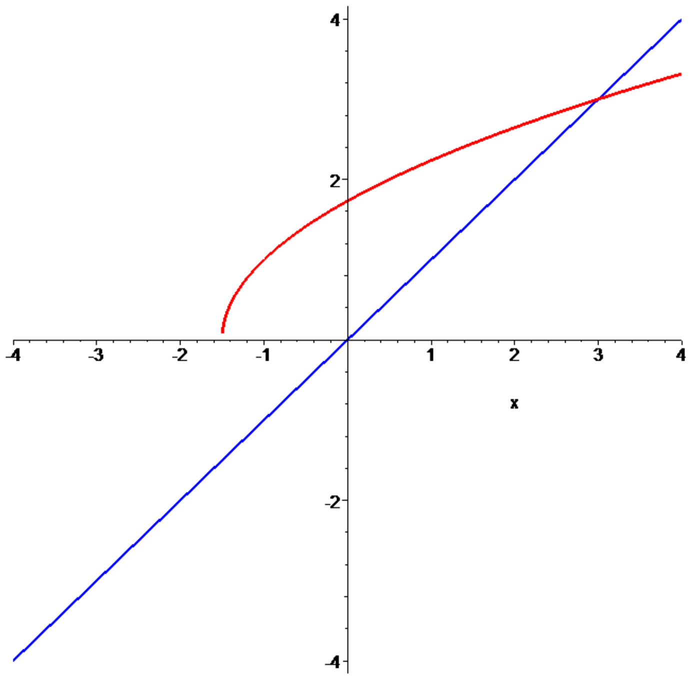
```
> factor(x^2-2*x-3);
```

$$(x+1)(x-3)$$

We pretend that we don't know this.

Suppose we rearrange to give this equivalent form:

$$x = g_1(x) = \sqrt{2x+3}$$



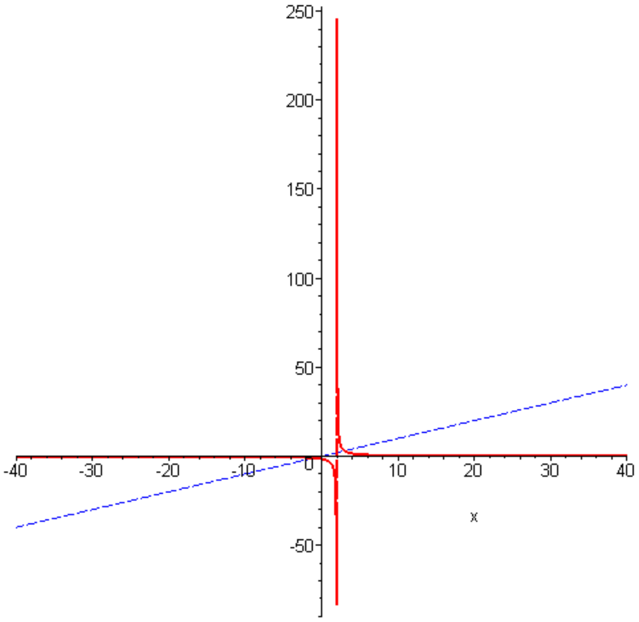
If we start with an initial value x=4 and iterate with the fixed point algorithm, successive values of x are

$$x_0 = 4,$$
$$x_1 = \sqrt{11} = 3.31662,$$
$$x_2 = \sqrt{9.63325} = 3.10375,$$
$$x_3 = \sqrt{9.20750} = 3.03439,$$
$$x_4 = \sqrt{9.06877} = 3.01144,$$
$$x_5 = \sqrt{9.02288} = 3.00381$$

and it appears that the values are converging on the root at x=3.

Other Rearrangements

Another rearrangement of $f(x)$ is $x = g_2(x) = \dfrac{3}{(x-2)}$



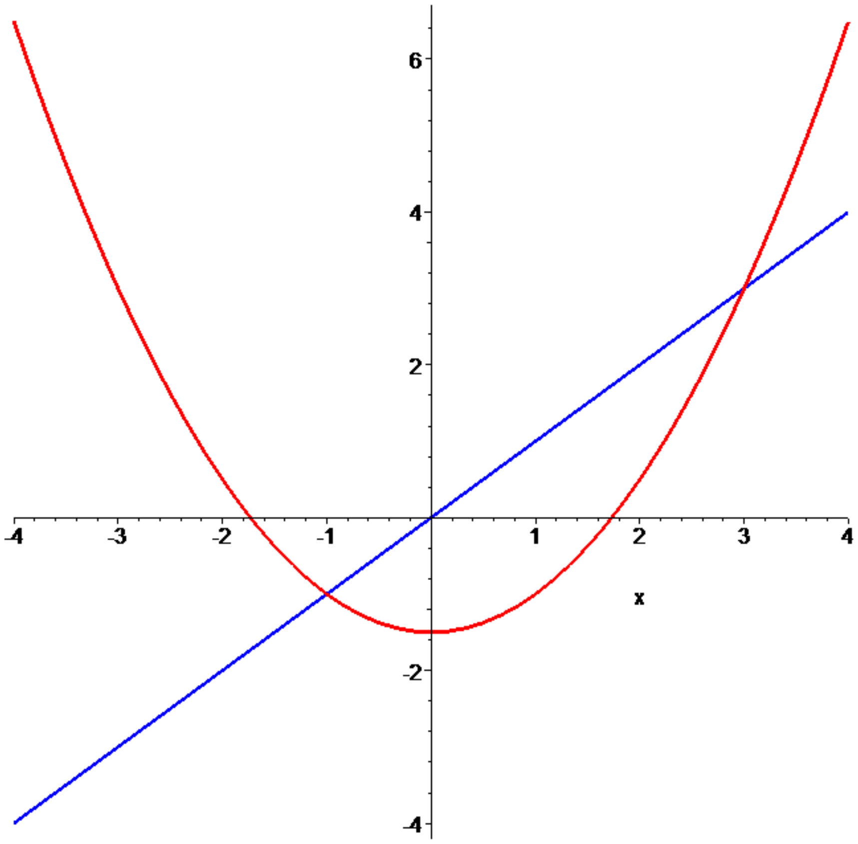
Let us start the iteration again with $x_0=4$, Successive values then are

$$
\begin{aligned}
x_0 &= 4, \\
x_1 &= 1.5, \\
x_2 &= -6, \\
x_3 &= -0.375, \\
x_4 &= -1.263158, \\
x_5 &= -0.919355 \\
x_6 &= -1.022762 \\
x_7 &= -0.990876 \\
x_5 &= -1.00305
\end{aligned}
$$

And it seems that we now converge to the other root, at $x = -1$.

Consider third rearrangement

$$x = g_3(x) = \frac{(x^2 - 3)}{2}$$



Let us start the iteration again with x_0=4, Successive values then are

$$x_0 = 4,$$
$$x_1 = 6.5,$$
$$x_2 = 19.625,$$
$$x_3 = 191.070$$

and the iterates are obviously diverging. (WHY?)

Convergence

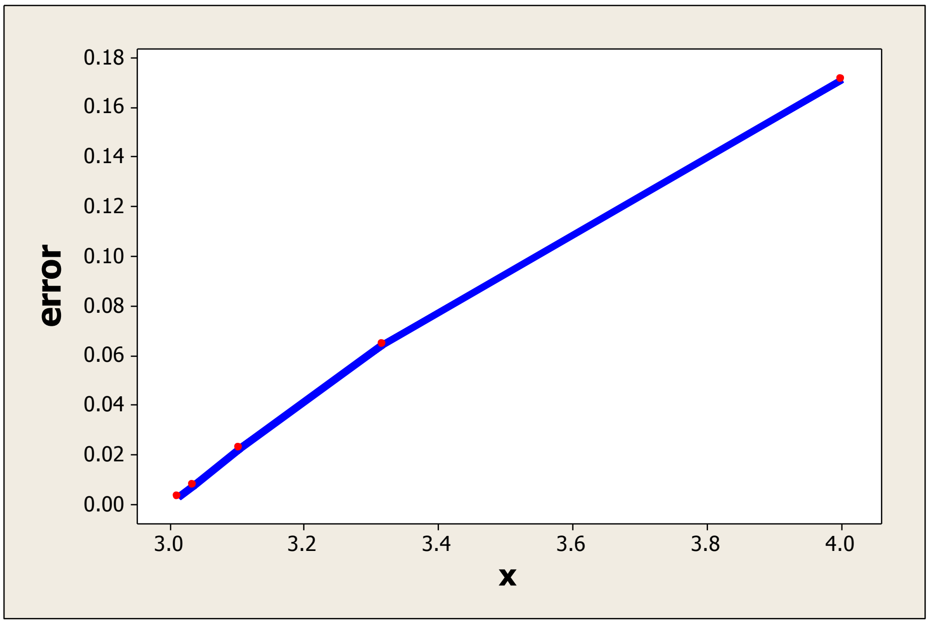
If $\left|g^{'}(x)\right| > 1$ **then the iteration** $x_{n+1} = g(x_n)$ **produces a sequence that** **diverges** **away from x.**

Order of Convergence

The fixed point method <u>converges at a linear rate</u>; it is said to be linearly convergent, meaning that <u>the error at each successive iteration is a constant fraction of the previous error.</u>

$$x = g_1(x) = \sqrt{2x+3}$$

Iteration	x_r	Absolute Relative Error
0	4.00000	*
1	3.31662	0.170845
2	3.10375	0.064183
3	3.03439	0.022347
4	3.01144	0.007563
5	3.00381	0.002534



Algorithm for Fixed Point Iteration

To determine a root of $f(x)=0$, given a value x_0 reasonably close to the root.

Rearrange the equation to an equivalent form $x=g(x)$.

Repeat

Set $\mathbf{x_1=x_0}$

Set $x_1=g(x_1)$

Until $|x_1-x_0)|<TOLERANCE$ **or** $|f(x_1)|<TOLERANCE$

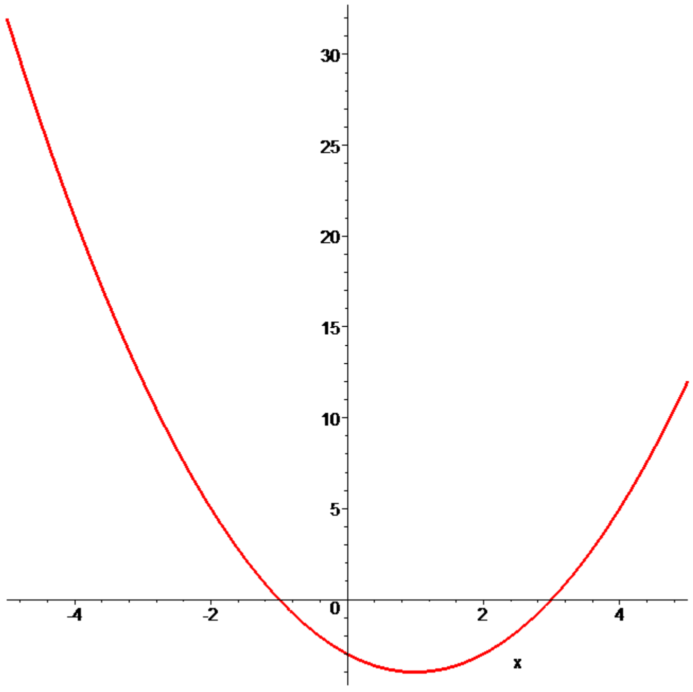
Note: The method *may converge* to a root different from the expected one, or it may diverge. Different rearrangements will converge at different rates.

MATLAB M-File (Fixed Point)

```
function [k,x,err,P] = fixedpoint(g,x0,tol,maxit)
%Input   -  g is the iteration function
%          -  x0 is the initial guess for the fixed-point
%          -  tol is the tolerance
%          -  maxit is the maximum number of iterations
%Output-  k is the number of iterations
%          -  x is the approximation to the fixed-point
%          - err is the error in the approximation
P(1)= x0;
for k=2:maxit
        P(k)=g(P(k-1));
        err=abs(P(k)-P(k-1));
        relerr=err/(abs(P(k))+eps);
        x=P(k);
    X=[k,x]
        if (err<tol) | (relerr<tol),break;end
end
if k==maxit
     disp('maximum number of iterations exceeded')
 end
```

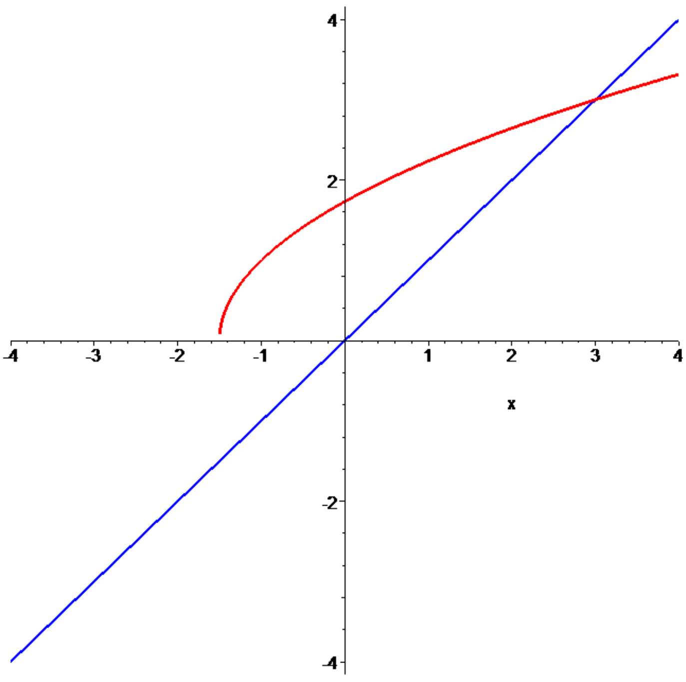
Example:

$$f(x) = x^2 - 2x - 3 = 0$$



Suppose we rearrange to give this equivalent form:

$$x = g_1(x) = \sqrt{2x+3}$$



If we start with an initial value x=4 and iterate with the fixed point algorithm, successive values of x are

```
>> g=inline('sqrt(2*x+3)')
g =
     Inline function:
     g(x) = sqrt(2*x+3)
>> fixedpoint(g,4,0.00001,20)
X =
    2.0000    3.3166
    3.0000    3.1037
    4.0000    3.0344
    5.0000    3.0114
    6.0000    3.0038
    7.0000    3.0013
    8.0000    3.0004
    9.0000    3.0001
   10.0000    3.0000
   11.0000    3.0000
   12.0000    3.0000
ans = 3.00
```

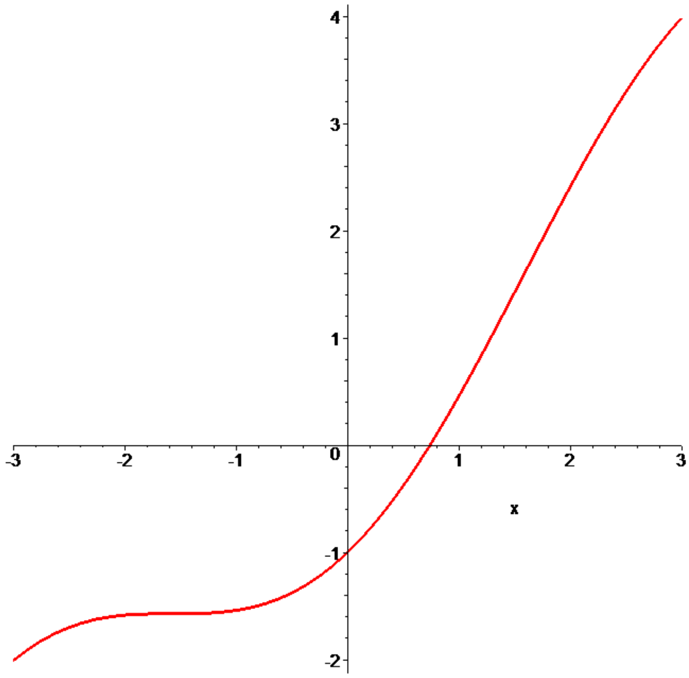
$$x = g_3(x) = \frac{(x^2 - 3)}{2}$$

```
>> g=inline('(x^2-3)/2')
g =
     Inline function:
     g(x) = (x^2-3)/2
>> fixedpoint(g,4,0.00001,20)
X =
    2.0000    6.5000
    3.0000   19.6250
    4.0000  191.0703
.
.
maximum number of iterations exceeded

ans =
```

Example:

$$f(x) = x - \cos(x)$$



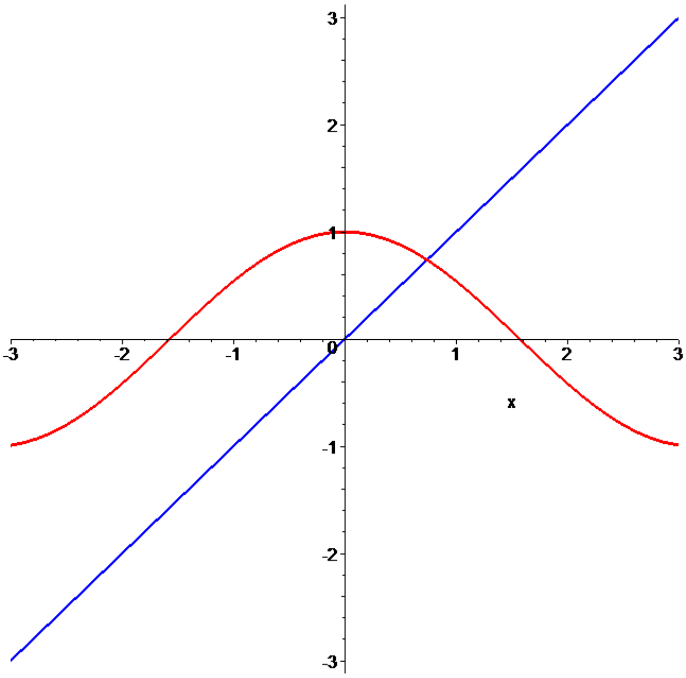
We shall consider three rearrangement of $x - \cos(x) = 0$

$$x = g_1(x) = \cos(x)$$

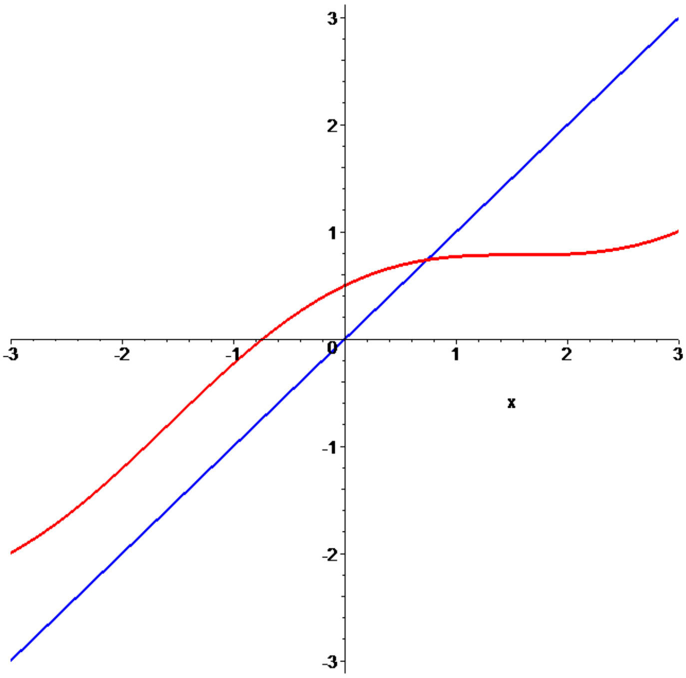
$$x = g_2(x) = x - \frac{x - \cos(x)}{2}$$

$$x = g_3(x) = x + \frac{x - \cos(x)}{2}$$

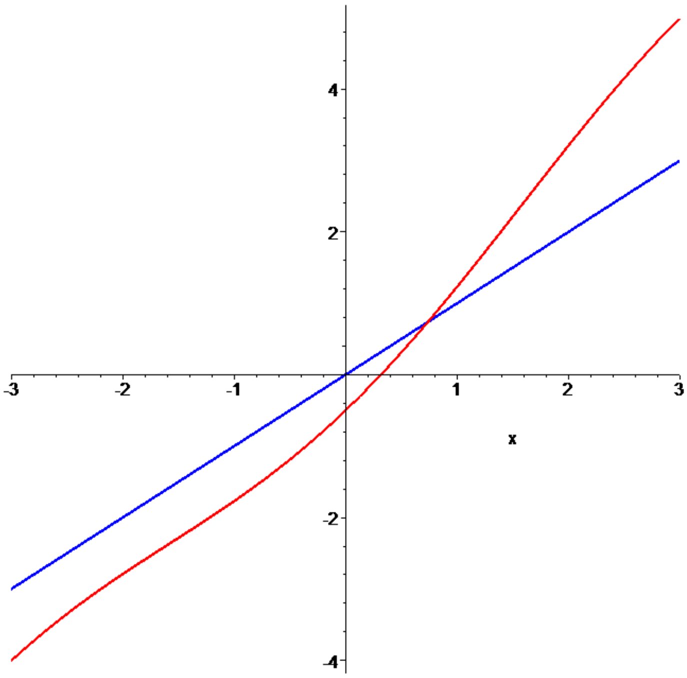
$$x = g_1(x) = \cos(x)$$

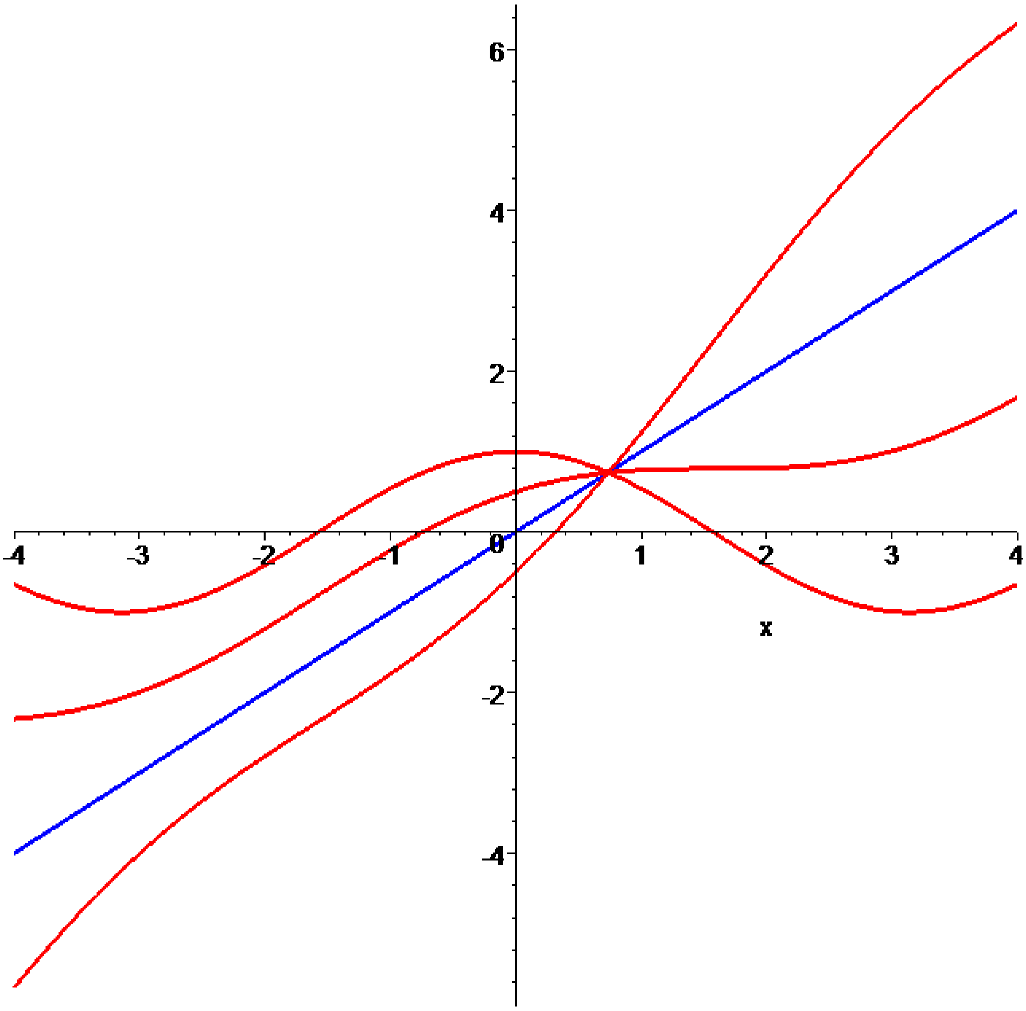


$$x = g_2(x) = x - \frac{x - \cos(x)}{2}$$



$$x = g_3(x) = x + \frac{x - \cos(x)}{2}$$




6
4
2
-2
-4
-4
-3
-2
-1
0
1
2
3
4
x

MATLAB Solution

$$x = g_1(x) = \cos(x)$$

```
>> g=inline('cos(x)')
g =
     Inline function:
     g(x) = cos(x)
>> fixedpoint(g,1.0,0.00001,20)
X =
    2.0000    0.5403
    3.0000    0.8576
    4.0000    0.6543
    5.0000    0.7935
    6.0000    0.7014
    7.0000    0.7640
    8.0000    0.7221
    9.0000    0.7504
   10.0000    0.7314
   11.0000    0.7442
   12.0000    0.7356
   13.0000    0.7414
   14.0000    0.7375
   15.0000    0.7401
```

MATLAB Solution

$$x = g_2(x) = x - \frac{x - \cos(x)}{2}$$

```
>> g=inline('x-0.5*(x-cos(x))')
g =
     Inline function:
     g(x) = x-0.5*(x-cos(x))
>> fixedpoint(g,0.50,0.001,20)
X =
    2.0000    0.6888
    3.0000    0.7304
    4.0000    0.7377
    5.0000    0.7389
    6.0000    0.7390
```

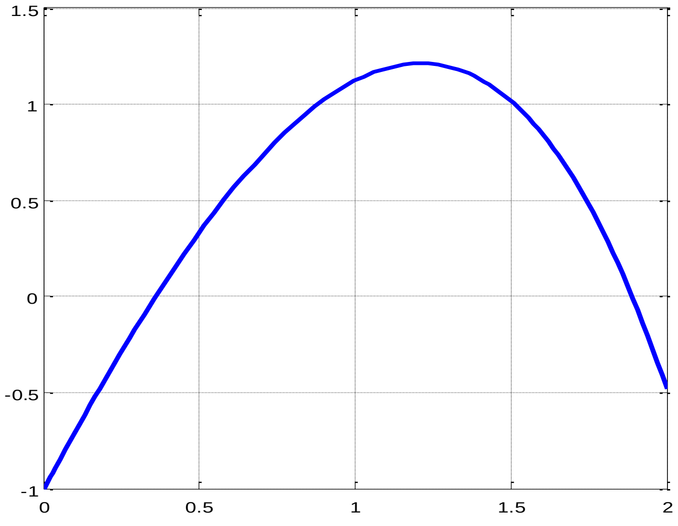
MATLAB Solution

$$x = g_3(x) = x + \frac{x - \cos(x)}{2}$$

```
>> g=inline('x+0.5*(x-cos(x))')
g =
     Inline function:
     g(x) = x+0.5*(x-cos(x))
>> fixedpoint(g,0.50,0.001,20)
X =
    2.0000    0.3112
    3.0000   -0.0092
    4.0000   -0.5137
    5.0000   -1.2061
    6.0000   -1.9874
    7.0000   -2.7788
    8.0000   -3.7008
    9.0000   -5.1273
   10.0000   -7.8925
.
   17.0000 -136.1280
   18.0000 -203.9387
   19.0000 -305.4254
   20.0000 -457.7528
maximum number of iterations exceeded
```

Example:

$$f(x) = 3x + \sin(x) - e^x$$

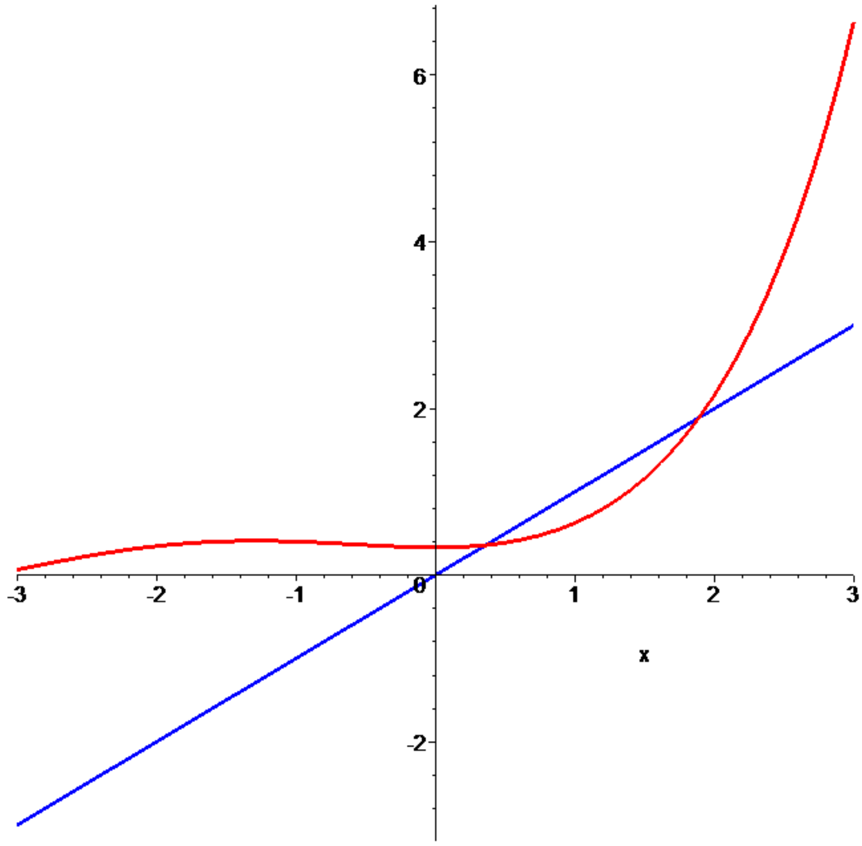


Iteration functions

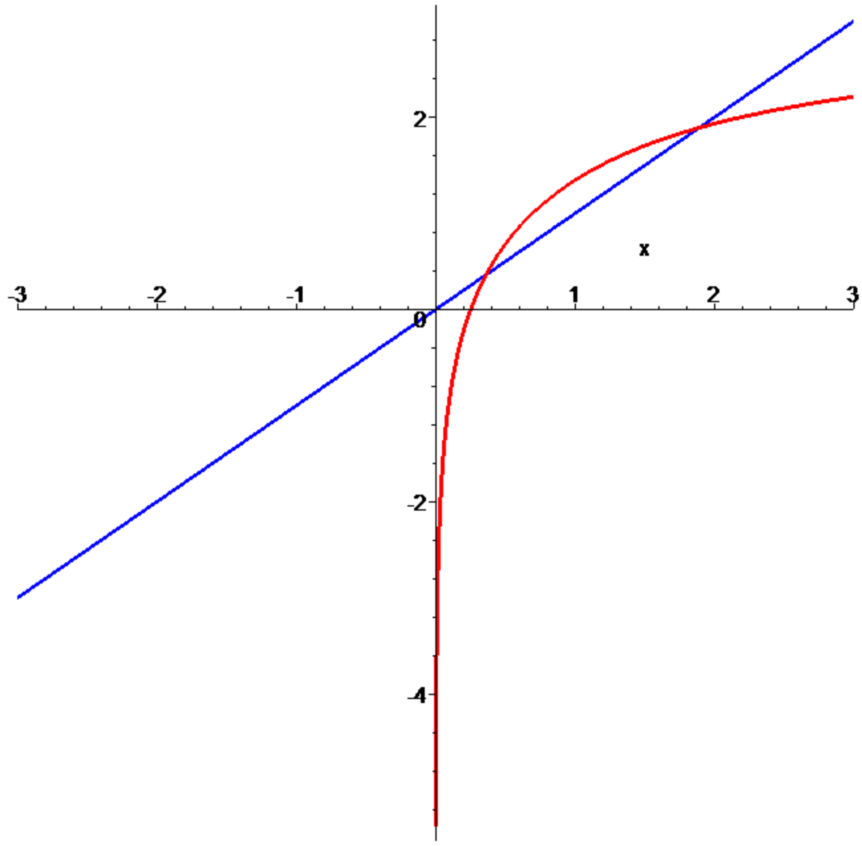
$$g_1(x) = \frac{e^x - \sin(x)}{3}$$

$$g_2(x) = \log(3x + \sin(x))$$

$$g_1(x) = \frac{e^x - \sin(x)}{3}$$



$$g_2(x) = \log(3x + \sin(x))$$



```
g1=inline('(exp(x)-sin(x))/3')
g1 =
     Inline function:
     g1(x) = (exp(x)-sin(x))/3
>> fixedpoint(g1,0.5,0.0001,10)
X =
    2.0000    0.3898
    3.0000    0.3656
    4.0000    0.3613
    5.0000    0.3606
    6.0000    0.3604
    7.0000    0.3604


>> g2=inline('log(3*x+sin(x))')
g2 =
     Inline function:
     g2(x) = log(3*x+sin(x))

>> fixedpoint(g2,0.5,0.0001,10)

X =    2.0000    0.6828
       3.0000    0.9856
       4.0000    1.3325
       5.0000    1.6032
       6.0000    1.7594
       7.0000    1.8343
       8.0000    1.8669
       9.0000    1.8806
      10.0000    1.8862
maximum number of iterations exceeded
```

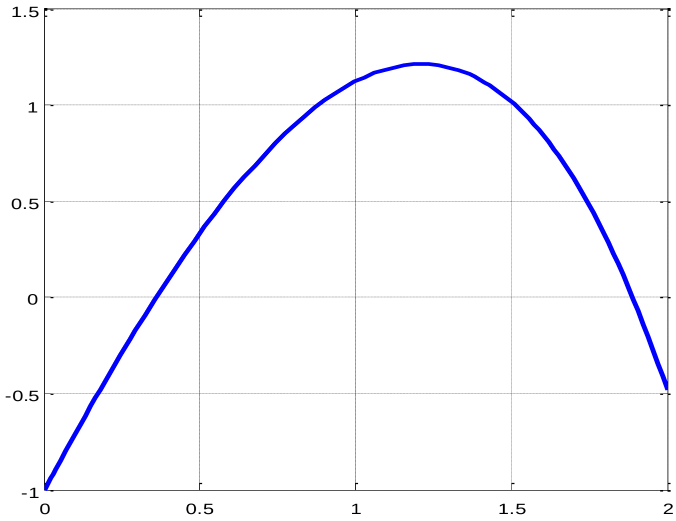
If we choose x_0=2.5 for both cases

```
g1inline('(exp(x)-sin(x))/3')
g1 =
     Inline function:
     g1(x) = (exp(x)-sin(x))/3
>> fixedpoint(g1,2.5,0.0001,10)
X =
    2.0000    3.8613
    3.0000   16.0627
    4.0000 1.0e+006 *
     5            Inf
     6           NaN
maximum number of iterations exceeded
```

Method diverges. WHY?

```
>> g2=inline('log(3*x+sin(x))')
g2 =
     Inline function:
     g2(x) = log(3*x+sin(x))
>> fixedpoint(g2,2.5,0.0001,10)
X =
    2.0000    2.0917
    3.0000    1.9661
    4.0000    1.9200
    5.0000    1.9021
    6.0000    1.8949
    7.0000    1.8920
    8.0000    1.8908
    9.0000    1.8904
   10.0000    1.8902
```

Approximates to the second root shown in the figure



Fixed Point Iteration Program for x = g(x)

Enter expression g(x):	3.83*x*(1-x
Enter initial x:	0.2
Starting iteration number for display:	0
Ending iteration number for display:	15

x = 3.83*x*(1-x)	
n	x
0	0.2
1	0.6128
2	0.9087676928
3	0.3175413678269552
4	0.829948860994241
5	0.5404259268177138
6	0.9512408012087574
7	0.17764206161275328
8	0.5595069271099131
9	0.943937685147333
10	0.20268104043408505
11	0.6189335009625182
12	0.9033239695958989
13	0.3344730403542266
14	0.8525611621645335
15	0.4814334011541312