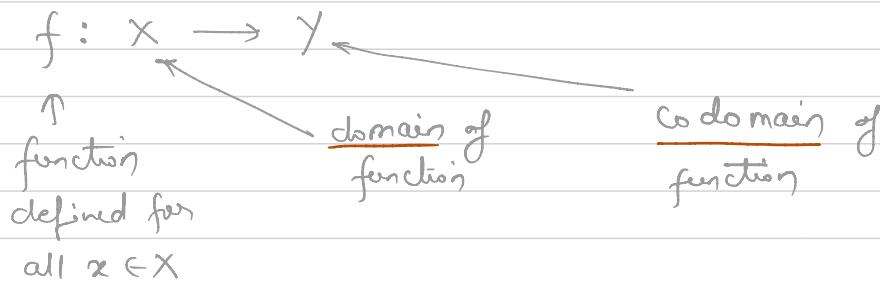Recap

(1.) Complete definition of a function includes the function itself, domain of function, and set in which function values lie.

$$f : X \longrightarrow Y$$

↑
function
defined for
all $x \in X$

domain of
function

Codomain of
function

Examples

(i) $f_1(x) = x^2$, $X = (-\infty, \infty)$, $Y = (-\infty, \infty)$

(ii) $f_2(x) = x^2$, $X = (-\infty, \infty)$, $Y = [0, \infty)$

(iii) $f_3(x) = x^2$, $X = [0, \infty)$, $Y = (-\infty, \infty)$

(iv) $f_4(x) = x^2$, $X = [0, \infty)$, $Y = [0, \infty)$

(v) $f_5(x) = \cos x$, $X = (-\infty, \infty)$, $Y = (-\infty, \infty)$

(vi) $f_6(x) = \cos x$, $X = (-\infty, \infty)$, $Y = [-1, 1]$

(vii) $f_7(x) = \cos^2 x$, $X = (-\infty, \infty)$, $Y = [0, 1]$

(2.) Domain, Codomain, range of a function $f : X \longrightarrow Y$

Dom(f) =   set of all x for which f is defined = X

Codom(f) =   set in which all possible values of f lie = Y

Rg(f) =   set of $f(x)$ for all $x \in X$

We have   Rg(f) $\subset$ Codom(f)

↑
subset

Examples of Range & Codomain

— for functions f_1 & f_2

$$Rg(f) = [0, \infty) \quad \subset \quad Codom(f) = (-\infty, \infty)$$
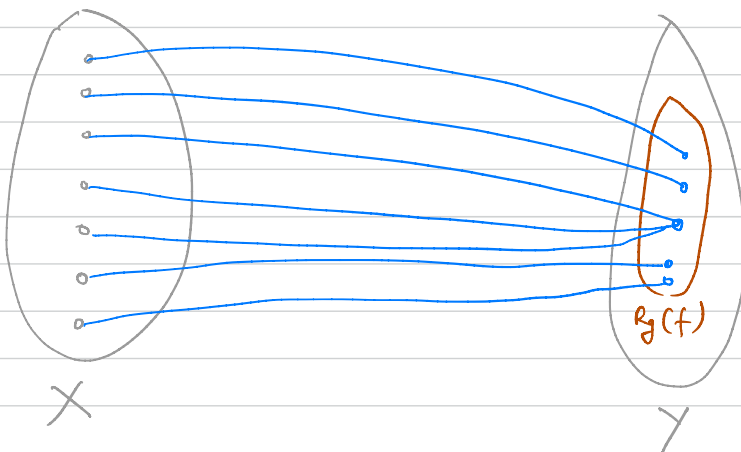
— for functions f_2 & f_4

$$Rg(f) = [0, \infty) \quad = \quad Codom(f) = [0, \infty)$$

— for function f_5, f_6, f_7

$$Rg(f_5) = [-1, 1] \quad \subset \quad Codom(f_5) = (-\infty, \infty)$$

$$Rg(f_6) = [-1, 1] \quad = \quad Codom(f_6) = [-1, 1]$$

$$Rg(f_7) = [0, 1] \quad = \quad Codom(f_7) = [0, 1]$$

X

Y

$Rg(f)$
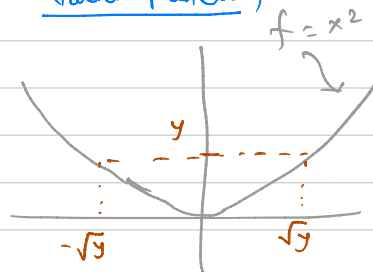
① Two different $x_1, x_2 \in X$ can have $f(x_1) = f(x_2)$

② But one point $x \in X$ can not have two values of $f(x)$

Valid Function

$f = x^2$

y

$-\sqrt{y}$ $\sqrt{y}$

for $x_1 = -\sqrt{y}, x_2 = \sqrt{y}$

$f(x_1) = f(x_2) = y$

Not a function

f

x

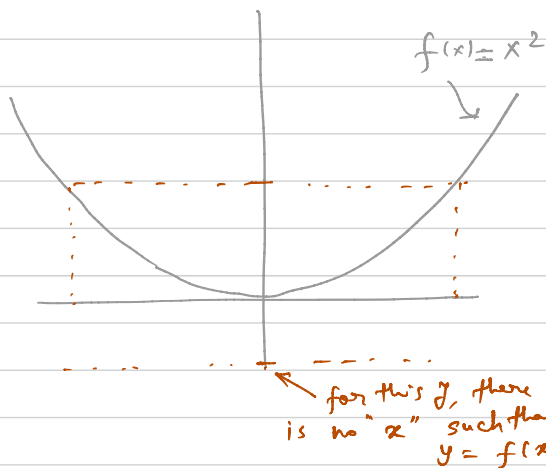x is mapped to two different points on circle

(3.) Types of functions

Surjective: $f: X \longrightarrow Y$ is surjective if for any
$y \in Y$, there is a $x \in X$ such that
$$y = f(x)$$

$f(x) = x^2$

(Case 1) $f(x) = x^2$, $X = (-\infty, \infty)$
$\qquad\qquad\qquad\qquad Y = (-\infty, \infty)$

"not surjective"

(Case 2) $f(x) = x^2$, $X = (-\infty, \infty)$
$\qquad\qquad\qquad\qquad Y = [0, \infty)$

for this y, there
is no "x" such that
$y = f(x)$

"surjective"

What changed from Case 1 to Case 2 ?
$\Downarrow$
Codom (f)

injective: $f: X \longrightarrow Y$ is injective if for any
$y \in Rg(f)$, there is a unique $x \in X$ such that
$$y = f(x)$$

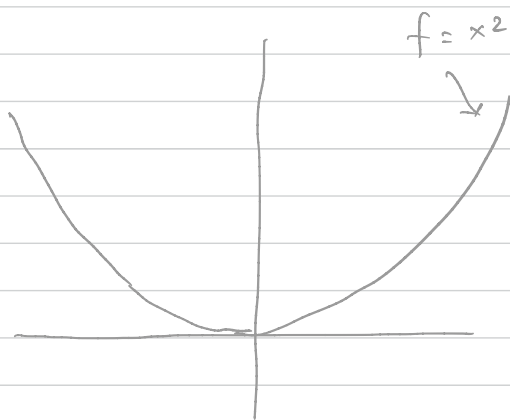Compare definition of surjective and injective
(i) in surjective, we have "for any $y \in Y$"
in injective, we have "for any $y \in Rg(f)$"

(ii) in surjective, we have "there is a $x \in X$"
in injective, we have "there is a unique $x \in X$"

$f = x^2$

(Case 1) $f = x^2$, $X = (-\infty, \infty)$
$Y = (-\infty, \infty)$

"not injective",   "not surjective"

(Case 2) $f = x^2$, $X = [0, \infty)$
$Y = (-\infty, \infty)$

"injective",   "not surjective"

(Case 3) $f = x^2$, $X = [0, \infty)$
$Y = [0, \infty)$

"injective",   "surjective"

all that we changed
is either
dom (f) or codom(f) !!

(4.) Closed and open intervals

(a, b)   open on both sides

$x \in (a, b) \Rightarrow a < x < b$

— x cannot be equal to a but it
can get as close to a as you
want

— x can not be equal to b but

it can get as close to b as
possible

[a, b]   open on left side

[a, b)   open on right side

[a, b]   closed on both sides, $a \le x \le b$

It matters in many situations whether the interval is open or closed

Example: $f: X \longrightarrow Y, \quad f(x) = x^2$

Consider a problem: find $x \in X$ such that $f(x) = 0$

Case 1: $X = (-\infty, \infty)$ Solution: $x = 0 \in (-\infty, \infty)$

Case 2: $X = (0, \infty)$ solution: there is no $x \in X$ such that $f(x) = 0$

$\downarrow$

"no solution"

Case 3: $X = [0, \infty)$ solution: $x = 0 \in X$

Consider another problem: find $x \in X$ such that

$$f(x) \leq f(z) \quad \text{for any}$$
$$z \in X$$

I.e. find $x \in X$ at which $f(x)$ is minimum

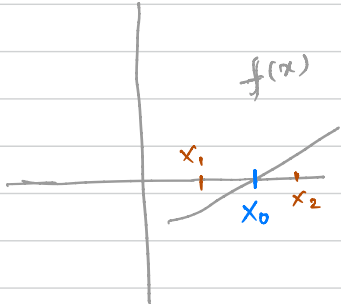Case 1: $X = [0, \infty)$ solution $x = 0 \in X$

$$f(x) = 0 \leq f(z) = z^2$$
$$\text{for any } z \in X$$

Case 2: $X = (0, \infty)$ "no solution"

Roots problem: Given a function $f : X \longrightarrow Y$, find $x \in X$
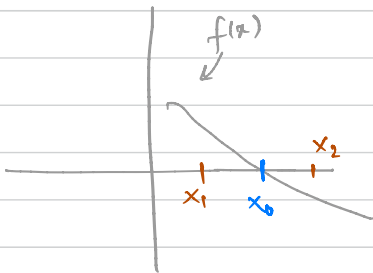
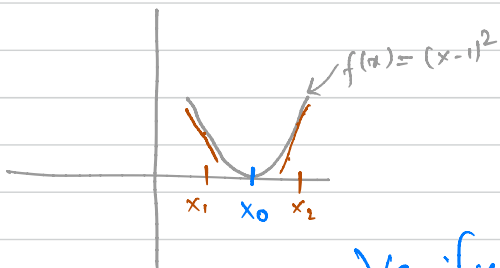such that $f(x) = 0$

Case 1

$f(x)$

x_1

x_0 x_2

$f(x_1) < 0, \qquad f(x_2) > 0$

$$f(x_1) \, f(x_2) \; < 0$$

$\Longleftarrow$ for any functions of Case 1 or Case 2 type

Case 2

$f(x)$

x_2

x_1 x_0

$f(x_1) > 0, \qquad f(x_2) < 0$

Case 3

$f(x) = (x-1)^2$
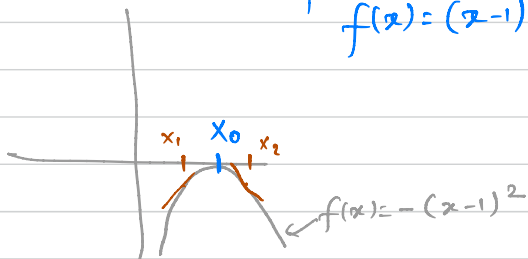
x_1 x_0 x_2

$f(x_1) > 0, \qquad f(x_2) > 0$

$f'(x_1) < 0, \qquad f'(x_2) > 0$

$$f'(x_1) \, f'(x_2) \; < 0$$

$\Longleftarrow$ for any function of Case 3 & 4

Verify $\Longrightarrow$

for $f(x) = (x-1)^2$!

Case 4

x_1 x_0 x_2

$f(x) = -(x-1)^2$

$f(x_1) < 0, \qquad f(x_2) < 0$

$f'(x_1) > 0, \qquad f'(x_2) < 0$

Graphical method

We already used graphical method in analyzing Case 1, 2, 3, 4 in previous page.

— It is extremely powerful in analyzing the properties of a function near point x_0 such that $f(x_0) = 0$

— The properties we observed will be used in developing numerical method

— But it is inefficient, can not be automated, and accuracy is limited

Bracketing method

We considers problems with function that fall in either case 1 or Case 2

Given two points $x_1, x_2 \in X$ with $x_1 < x_2$
    and $f(x_1) f(x_2) < 0$

then there exist $x_0 \in [x_1, x_2]$ such that

$$f(x_0) = 0$$

Bracketing methods use above property to locate point x_0
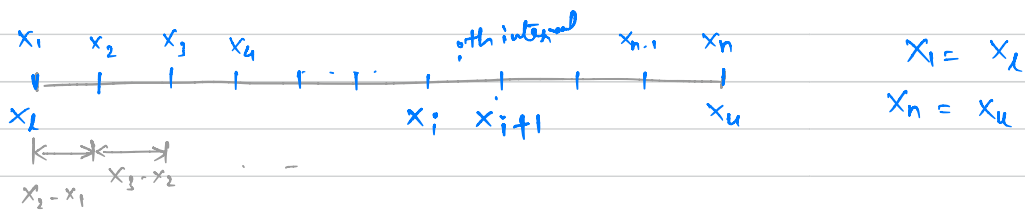
There are several bracketing methods

(i) incremental search method
(ii) bisection method
(iii) false position method

Incremental search method
Let $f: X \to Y = (-\infty, \infty)$ be a function.

step 0: Suppose we know two numbers $x_u, x_l \in X$ such that
$$f(x_u) f(x_l) < 0 \implies x_0 \in [x_l, x_u]$$

step 1: Divide $[a, b]$ in smaller uniformly sized intervals

$$x_1 = x_l$$
$$x_n = x_u$$

Such that $\quad x_2 - x_1 = x_3 - x_2 = \cdots = x_n - x_{n-1}$

step 2: For each $i = 1, 2, \ldots, n-1$

    check $\quad f(x_i) f(x_{i+1}) < 0$

    $\downarrow$

If true then because $x_0 \in [x_i, x_{i+1}]$ such that $f(x_0) = 0$

we store $\dfrac{x_i + x_{i+1}}{2}$

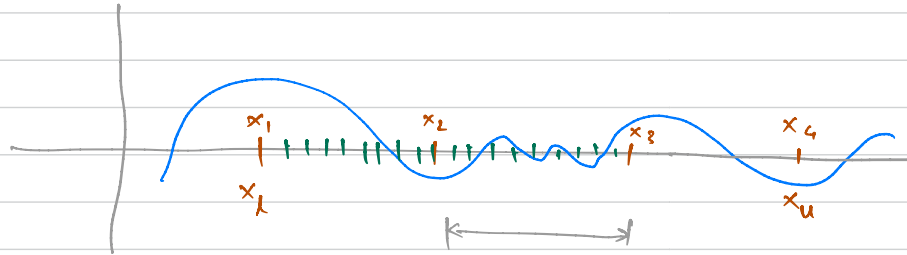or one of the solution (we treat midpoint as solution!)

At the end, we have list of $\dfrac{x_i + x_{i+1}}{2}$ for those

    i that satisfy $f(x_i) f(x_{i+1}) < 0$.

    $\Downarrow$

    our solution

Limitations of incremental search

(i) For cases where multiple x_0 exists and they are close by

in this interval there are 5 x_0 such that $f(x_0) = 0$ but the method will return only one solution from this interval $[x_2, x_3]$

Remedy : Take intervals of smaller size

Controlling errors in incremental search method

By taking intervals of smaller and smaller size, we can get arbitrarily close to all the point x_0 such that $f(x_0) = 0$

$\Downarrow$

we will look at errors more in next method

```matlab
function xb = incsearch(func,xmin,xmax,ns)
% incsearch: incremental search root locator
% syntax:
%     xb = incsearch(func,xmin,xmax,ns)
% input:
%     func = name of function
%     xmin, xmax = endpoints of interval
%     ns = number of subintervals (default = 50)
% output:
%     xb(k,1) is the lower bound of the kth sign change
%     xb(k,2) is the upper bound of the kth sign change
%     If no brackets found, xb = [].

% check if sufficient arguments are supplied to this function
if nargin < 3
    error('at least 3 arguments required')
end

%if ns blank set to 50
if nargin < 4
    ns = 50;
end

% Incremental search
x = linspace(xmin,xmax,ns);

% get all values of function
f = func(x);

% start search
nb = 0; xb = []; %xb is null unless sign change detected
for k = 1:length(x)-1
    %check for sign change
    if sign(f(k)) ~= sign(f(k+1))
        nb = nb + 1;
        xb(nb,1) = x(k);
        xb(nb,2) = x(k+1);
    end
end
```

Handwritten annotations:

$$\frac{X_{max} - X_{min}}{ns}$$

$X_{min}, \ X_{min} + \Delta, \ \ldots \ , \ X_{min} + n\Delta$

$$\Delta = \frac{X_{max} - X_{min}}{ns}$$

$\Rightarrow \ func(x(k)) * func(x(k+1)) < 0$

$$xb = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}$$

```matlab
1    % function
2    f = @(x) sin(10*x)+cos(3*x);
3
4    % lower and upper limits of arguments
5    xl = 3;
6    xu = 6;
7    x = xl:(xu-xl)/100:xu;
8    y = f(x);
9
10   % get intervals containing roots
11   n_intervals_1 = 50;
12   xb1 = incsearch(f, xl, xu, n_intervals_1);
13   disp('intervals containing roots')
14   disp(xb1)
15
16   % increase number of intervals (take smaller width intervals)
17   n_intervals_2 = 100;
18   xb2 = incsearch(f, xl, xu, n_intervals_2);
19   disp(' ')
20   disp('intervals containing roots')
21   disp(xb2)
22
23   % plotting
24   figure(1)
25   subplot(2, 1, 1)
26   plot(x, y, 'DisplayName', 'f(x)', 'LineWidth', 2)
27   grid on
28   hold on
29
30   for i=1:length(xb1)
31       a = xb1(i, 1);
32       b = xb1(i, 2);
33       rectangle('Position', [a, -0.1, b-a, 0.2], 'LineWidth', 2, ...
34           'FaceColor', 'r')
35       hold on
36   end
37   legend()
38   title('plot of root intervals (case 1)')
39
40   subplot(2, 1, 2)
41   plot(x, y, 'DisplayName', 'f(x)', 'LineWidth', 2)
42   grid on
43   hold on
44
45   for i=1:length(xb2)
46       a = xb2(i, 1);
47       b = xb2(i, 2);
48       rectangle('Position', [a, -0.1, b-a, 0.2], 'LineWidth', 2, ...
49           'FaceColor', 'r')
50       hold on
51   end
52   legend()
53   title('plot of root intervals (case 2)')
```

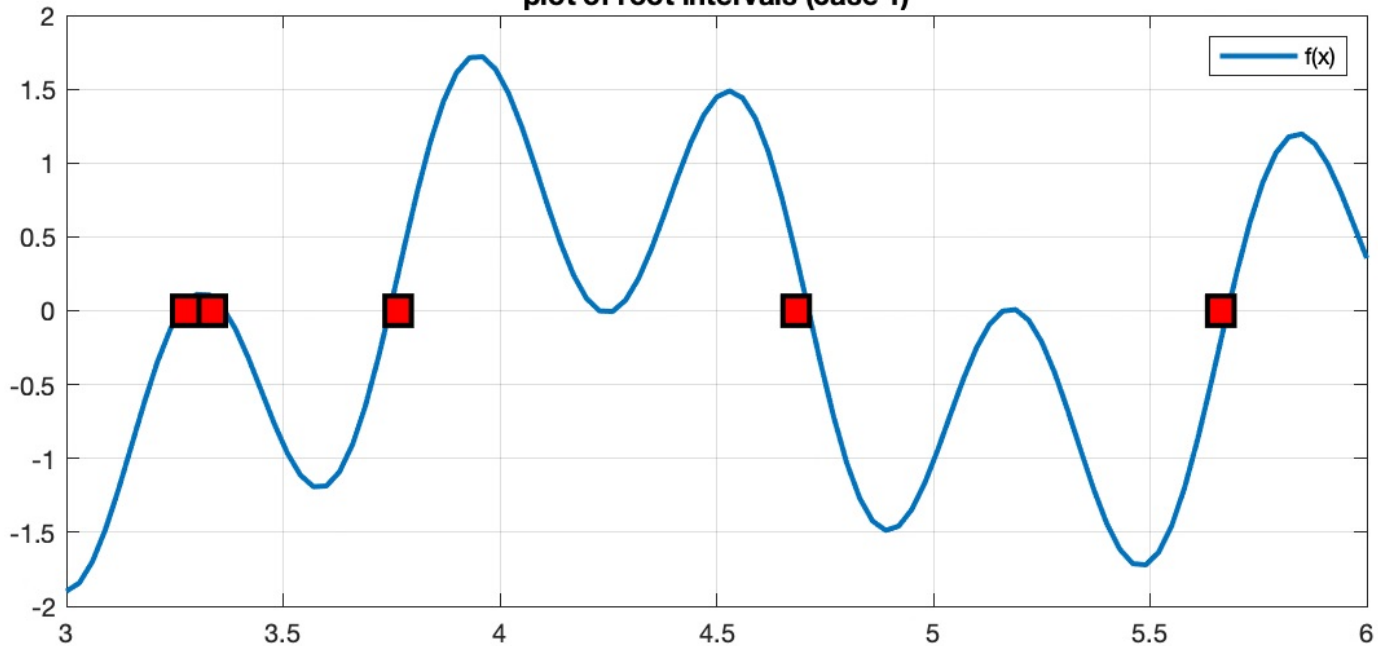$g = @(t) \ \sin(10*t) + \cos(3*t)$

$f(10) \quad , \quad g(10)$

$dfdx = @(x) \ 10*\cos(10*x)$
$\qquad\qquad -3*\sin(3*x)$

$h = @(x,y)$
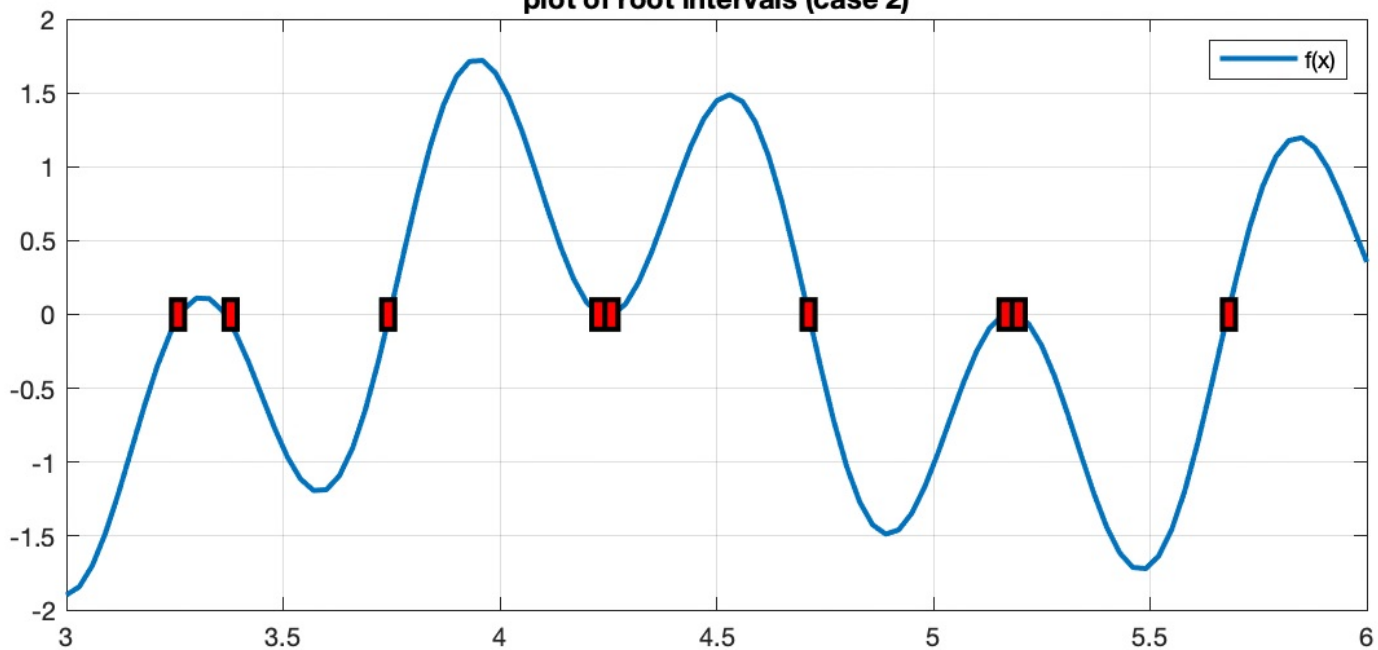$\qquad x^2 + y^2 - 2xy$

$h_{-1-1} = h(1,1)$

plot of root intervals (case 1)
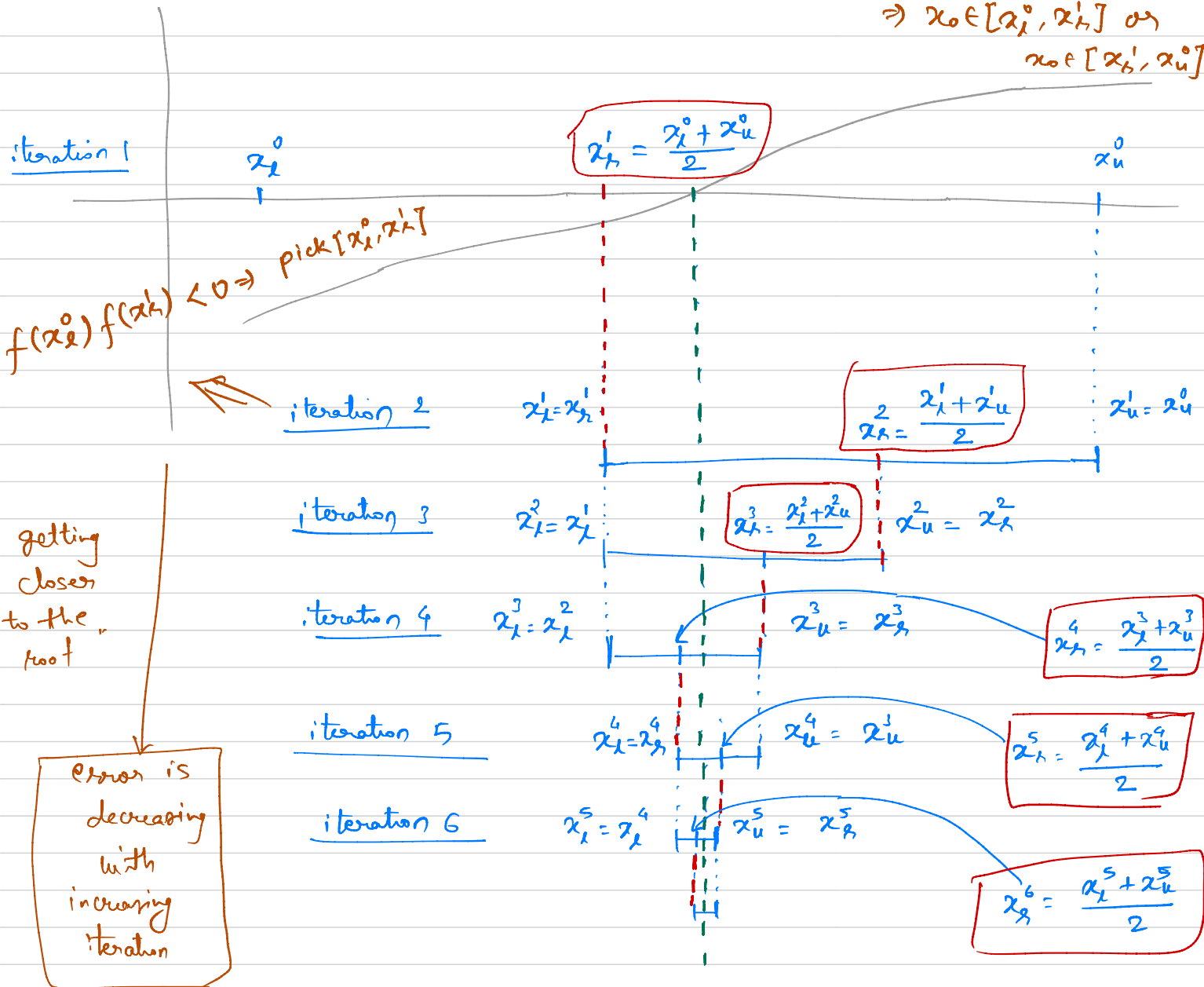

plot of root intervals (case 2)

Bisection method

This method allows calculation of roots more rapidly. However, it can only compute one root!

$$x_0 \in [x_l^0, x_u^0]$$

$$\Rightarrow x_0 \in [x_l^0, x_h^1] \text{ or}$$
$$x_0 \in [x_h^1, x_u^0]$$

iteration 1

x_l^0

$$x_h^1 = \frac{x_l^0 + x_u^0}{2}$$

x_u^0

$f(x_l^0) f(x_h^1) < 0 \Rightarrow$ pick $[x_l^0, x_h^1]$

iteration 2

$x_l^1 = x_l^1$

$$x_h^2 = \frac{x_l^1 + x_u^1}{2}$$

$x_u^1 = x_u^0$

iteration 3

$x_l^2 = x_l^1$

$$x_h^3 = \frac{x_l^2 + x_u^2}{2}$$

$x_u^2 = x_h^2$

iteration 4

$x_l^3 = x_l^2$

$x_u^3 = x_h^3$

$$x_h^4 = \frac{x_l^3 + x_u^3}{2}$$

iteration 5

$x_l^4 = x_h^4$

$x_u^4 = x_u^1$

$$x_h^5 = \frac{x_l^4 + x_u^4}{2}$$

iteration 6

$x_l^5 = x_l^4$

$x_u^5 = x_h^5$

$$x_h^6 = \frac{x_l^5 + x_u^5}{2}$$

"getting closer to the root"

error is decreasing with increasing iteration

But how is error **defined** ?

At the end of each iteration "i" bisection method produces improved solution x_h^i
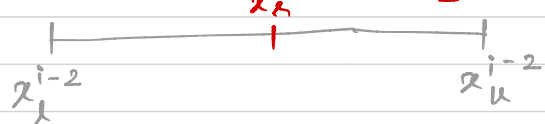
Error in bisection method

Let x_r^{i-1} root from iteration $i-1$ and x_l^{i-2} and x_u^{i-2}
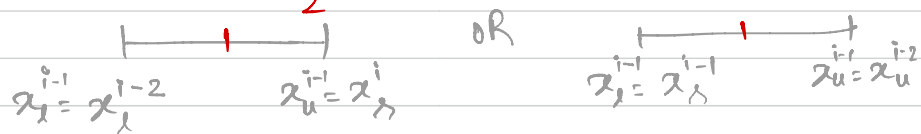
such that $x_r^{i-1} = \dfrac{x_l^{i-2} + x_u^{i-2}}{2}$

Let x_r^i root from iteration i and x_l^{i-1} and x_u^{i-1}

such that $x_r^i = \dfrac{x_l^{i-1} + x_u^{i-1}}{2}$

iter $i-1$

$$x_r^{i-1} = \dfrac{x_l^{i-2} + x_u^{i-2}}{2}$$

x_l^{i-2} x_u^{i-2}

iter i

$$x_r^i = \dfrac{x_l^{i-1} + x_l^{i-1}}{2} \qquad \text{OR} \qquad x_r^i = \dfrac{x_l^{i-1} + x_u^{i-1}}{2}$$

$x_l^{i-1} = x_l^{i-2}$ $x_u^{i-1} = x_r^i$ $x_l^{i-1} = x_r^i$ $x_u^{i-1} = x_u^{i-2}$

Thus

$$E_i := \left| x_r^i - x_r^{i-1} \right|$$

$$= \left| \dfrac{x_l^{i-1} + x_r^{i-1}}{2} - x_r^{i-1} \right| \qquad \text{or} \qquad \left| \dfrac{x_u^{i-1} + x_r^{i-1}}{2} - x_r^{i-1} \right|$$

$$\boxed{E_i = \left| \dfrac{x_l^{i-1} - x_r^{i-1}}{2} \right| \qquad \text{or} \qquad \left| \dfrac{x_u^{i-1} - x_r^{i-1}}{2} \right|}$$

$$= \left| \dfrac{x_l^{i-2} - \frac{1}{2}\left(x_l^{i-2} + x_u^{i-2}\right)}{2} \right| \qquad \text{or} \qquad \left| \dfrac{x_u^{i-2} - \frac{1}{2}\left(x_l^{i-2} + x_u^{i-2}\right)}{4} \right|$$

$$= \tfrac{1}{4} \left| x_u^{i-2} - x_l^{i-2} \right| \qquad \text{or} \qquad \tfrac{1}{4} \left| x_u^{i-2} - x_l^{i-2} \right|$$

Ment, note that $\rightarrow$ interval width in ith iteration $= \frac{1}{2^i}$ (initial interval width)

$$|x_u^i - x_l^i| = \frac{1}{2}|x_u^{i-1} - x_l^{i-1}| = \frac{1}{2^2}|x_u^{i-2} - x_l^{i-2}| \cdots$$

$$\longrightarrow = \frac{1}{2^i}|x_u^0 - x_l^0|$$

$\underline{\&}$

$$E_i = \frac{1}{2^2}|x_u^{i-2} - x_l^{i-2}| = \frac{1}{2^3}|x_u^{i-3} - x_l^{i-3}| = \frac{1}{2^4}|x_u^{i-4} - x_l^{i-4}|$$

$\rightarrow$ $\boxed{E_i = \frac{1}{2^i}|x_u^0 - x_l^0|}$ $\underline{where}$ x_l^0, x_u^0 are initial interval we started

$\underline{\&}$ error at iteration i is simply $\frac{1}{2^i}\Delta$ where

Δ is the size of initial interval

Normalized error in bisecting method $\longrightarrow$ $|x_l^i - x_l^{i-1}|$

$$e_a^i = \frac{E_a^i}{|x_l^0|} \times 100\%.$$

```matlab
function [xr, fxr, ea, iter] = bisect(func,xl,xu,ea_tol,maxit)
if nargin < 3
    error('at least 3 arguments required')
end
if nargin < 4
    maxit = 50;
end

xr = []; fxr = []; ea = [];
iter = 0; xl_new = xl; xu_new = xu; xr_new = 0; ea_new = 100;
while (1) % we terminate inside code
    iter = iter + 1;

    % reset old mid point
    xr_old = xr_new;
    xr_new = 0.5*(xl_new + xu_new);

    % set the new interval end points for next iteration
    xl_old = xl_new; xu_old = xu_new;

    % select either [xl_old, xr_new] or [xr_new, xu_old]
    f_product = func(xl_old)*func(xr_new);
    if f_product < 0 % left interval is selected
        xl_new = xl_old;
        xu_new = xr_new;
    else % right interval is selected
        xl_new = xr_new;
        xu_new = xu_old;
    end

    % compute error
    if iter > 1
        ea_new = abs(xr_new - xr_old) * 100 / abs(xr_new);
    end

    % save
    xr(iter) = xr_new; fxr(iter) = func(xr_new); ea(iter) = ea_new;

    % terminate
    if ea_new <= ea_tol || iter >= maxit
        break;
    end
end
end
```
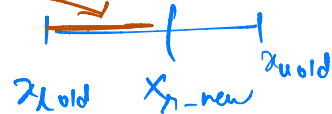
```matlab
% function
f = @(x) sin(10*x)+cos(3*x);

% lower and upper limits of arguments
xl = 3;
xu = 6;
x = xl:(xu-xl)/100:xu;
y = f(x);

% get root
maxit = 50;
ea_tol = 0.1;
[xr, fxr, ea, iter] = bisect(f, xl, xu, ea_tol, maxit);
disp('root')
disp(xr(end))

% plotting
ls = 4; ms = 10; ms2 = 20;
figure('DefaultAxesFontSize',20)
subplot(2, 1, 1)
plot(x, y, 'DisplayName', 'f(x)', 'LineWidth', ls)
grid on
hold on

% plot the solution at different iterations
plot(xr, fxr, 'ro', 'LineStyle', 'none', ...
    'DisplayName', 'xr(iteration)', 'MarkerSize', ms)
hold on
plot(xr(end), fxr(end), 'g+', ...
    'DisplayName', 'xr(final)', 'MarkerSize', ms2)
legend()
title('roots from different iterations')

subplot(2, 1, 2)
iter_i = 1:1:iter;
plot(iter_i, ea, '+-', 'DisplayName', 'error', ...
    'LineWidth', ls, 'MarkerSize', ms)
legend()
title('error in bisection method')
```
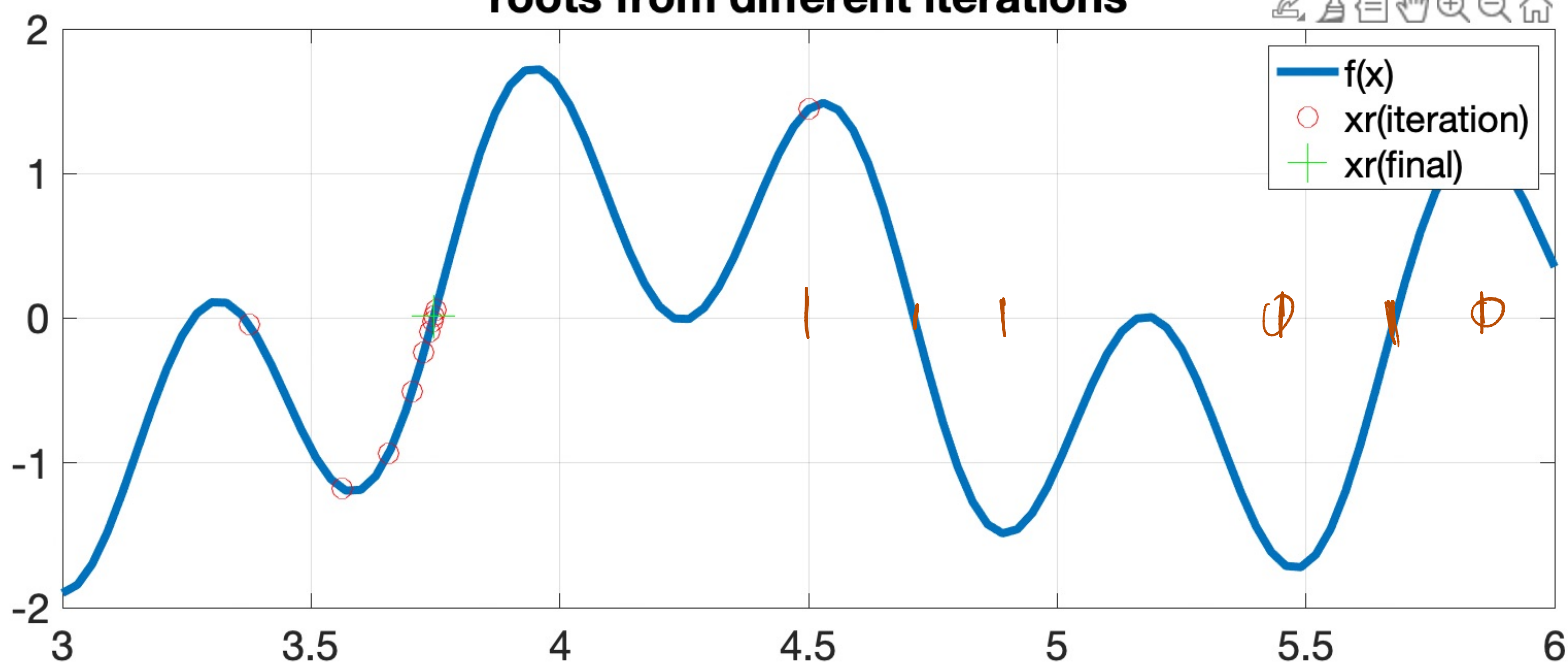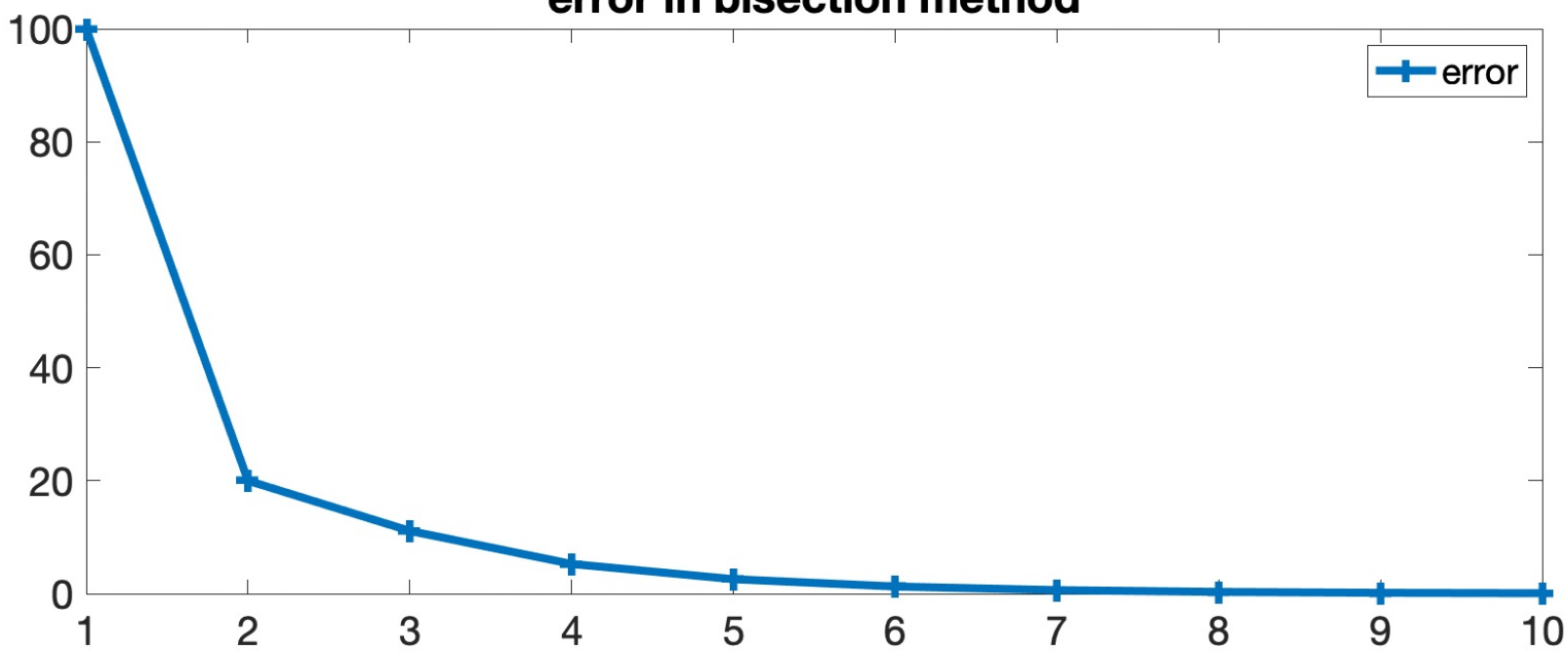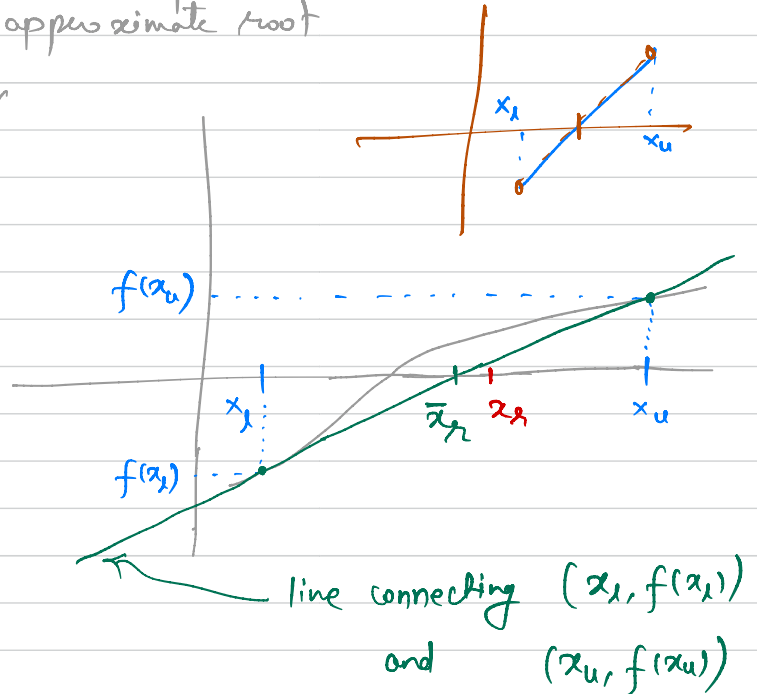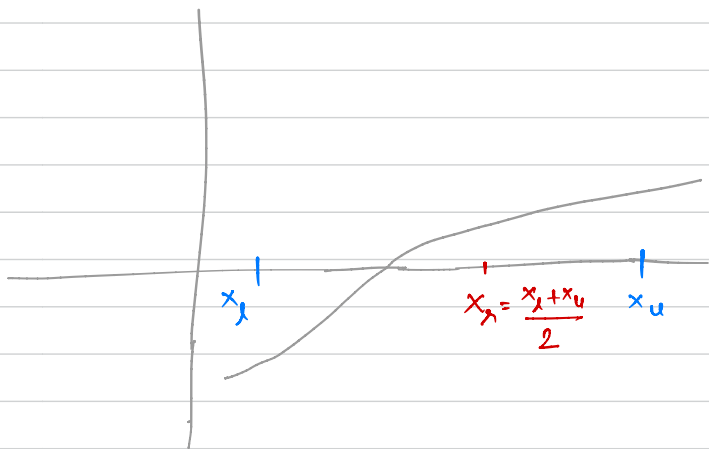
roots from different iterations

error in bisection method

The false-position method

In bisection method, we take the mid point of interval as approximate root. False-position method instead uses clever way to get the approximate root

$$\Downarrow$$

$$x_r = \frac{x_l + x_u}{2} \quad x_u$$

line connecting $(x_l, f(x_l))$
and $(x_u, f(x_u))$

☐ Consider a line connecting $(x_l, f(x_l))$ and $(x_u, f(x_u))$

Equation of this line is

We can also have
$$\Rightarrow \frac{(x_u) - y}{x_u - x} = \frac{f(x_u) - f(x_l)}{x_u - x_l}$$

$$y = f(x_u) - \frac{(f(x_u) - f(x_l))}{\left(\frac{x_u - x}{x_u - x_l}\right)}$$

$$\frac{y - f(x_l)}{x - x_l} = \frac{f(x_u) - f(x_l)}{x_u - x_l}$$

$$\boxed{y = f(x_l) + (f(x_u) - f(x_l))\left(\frac{x - x_l}{x_u - x_l}\right)}$$

☐ $\bar{x}_r$ such that $f = 0$

$$\Rightarrow f(x_l) + (f(x_u) - f(x_l))\left(\frac{\bar{x}_r - x_l}{x_u - x_l}\right) = 0$$

OR equivalently
$$\bar{x}_r = x_u - f(x_u)\left(\frac{f(x_u) - f(x_l)}{x_u - x_l}\right)$$

$$\Rightarrow \boxed{\bar{x}_r = x_l + \frac{(x_u - x_l)}{(f(x_u) - f(x_l))}(-f(x_l))}$$

$$\Downarrow$$

Compare this with bijection solution
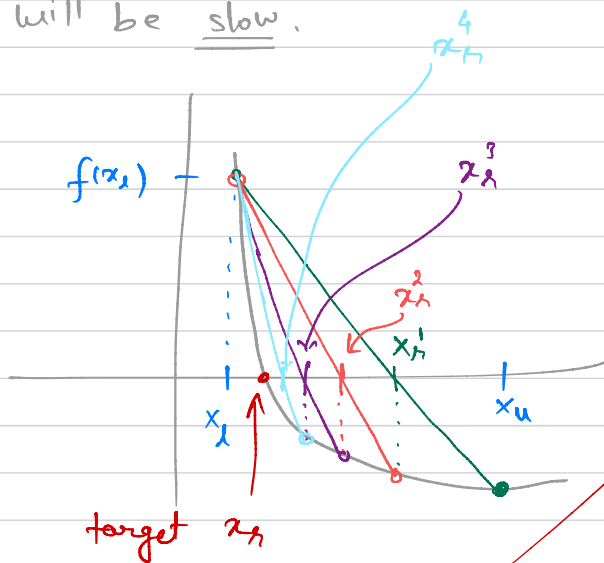
$$\boxed{x_r = \frac{x_l + x_u}{2}}$$

do false-position method uses values of function $f(x_l)$, $f(x_u)$ also to find the root $\bar{x}_r$.

Limitation of false-position method

for fuction with very large slope, the improvement in location of root after each iteration may not be substantial OR in other words decay of error with iteration will be <u>slow</u>.

$f(x_l)$ —

x_r^4

x_r^3

x_r^2

x_r^1

x_l x_u

target x_r

with each iteration,

(i) we are moving towards x_r

(ii) but the change from x_r^1 to x_r^2, x_r^2 to x_r^3, x_r^3 to x_r^4, ... is not large and the change is getting

(iii) the closer we get to x_r, the harder it becomes in further improving the solution

due to the fact that $f(x_l)$ is very large resulting in lines with very large slopes