```
>> x = 2; y = 4; z = x + y        \n   \t
       2      3          5
```

the addition of two no. x = 2 and y = 4 is z = 6

```
disp([' ', num2str(x), ' ', num2str(y), ' ', num2str(z)]);

fprintf('the additon of two no. x = format and y = format
        is z = format', x, y, z)
```

% W . d f/e/g/d

```
fprintf('..... x = %5.2f and y = %6.3f is z = %10.5f',
        x, y, z)
```

x = 2.00 ---- y = 3.000 ... z = 5.00000

```
>> x = 0:10;   y = x.^2;

>> Z = [x ; y]

fprintf('%4.2f %4.2f \n', Z )
```

x | y

$$\begin{matrix} 0 & 1 & 2 & 3 & 4 & \cdots & 10 \\ 0 & 1 & 4 & 9 & 16 & \cdots & \end{matrix}$$

```
0  0
1  1
2  4
3  9
...
```

fid = fopen('filename.dat/txt','w')

w →

r →

a

fprintf(fid, '   ', Z)

fclose(fid)

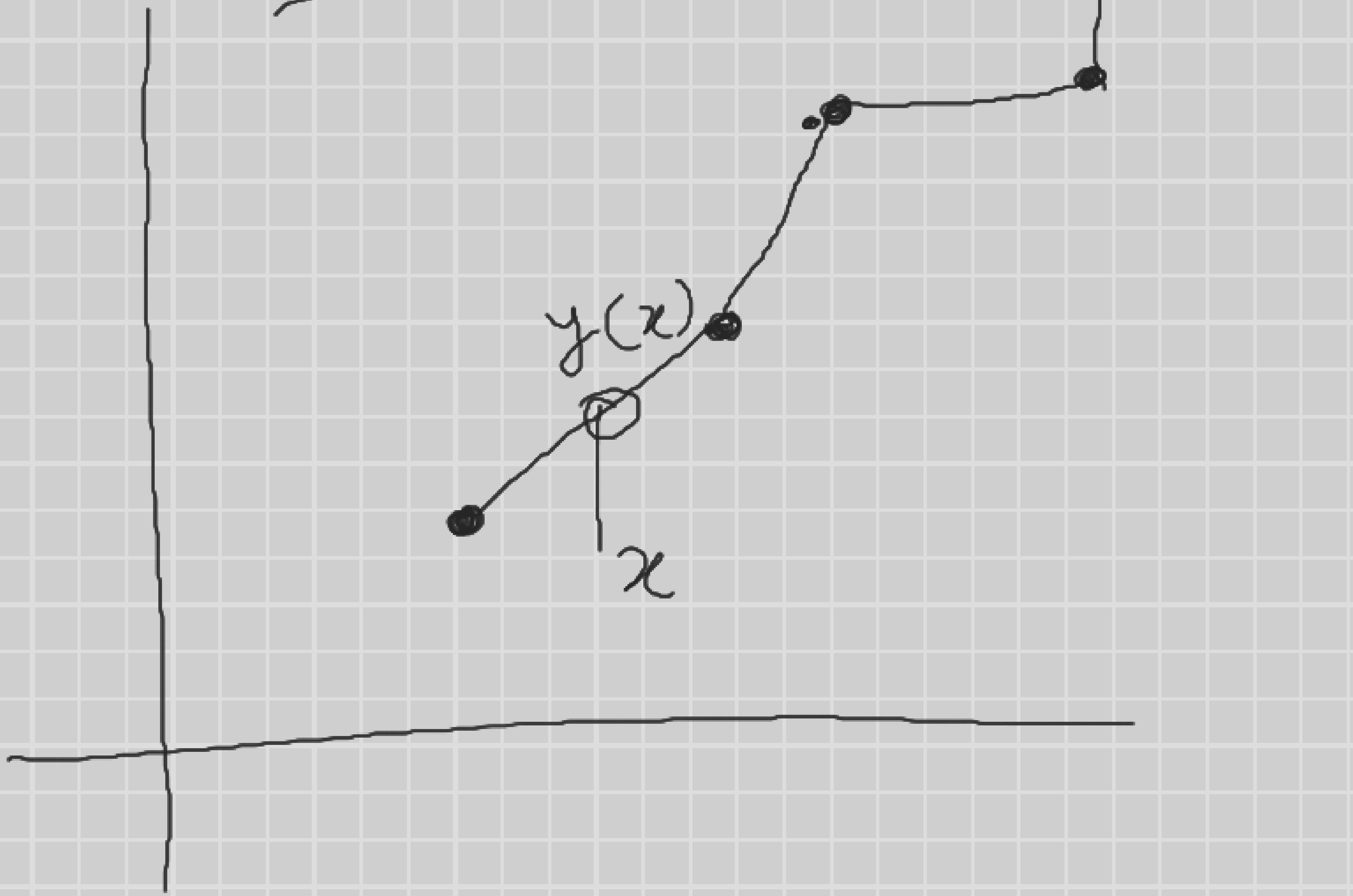
Interpolation

X	Y
x_1	y_1
x	y
x_2	y_2
x_3	y_3
⋮	
x_n	y_n

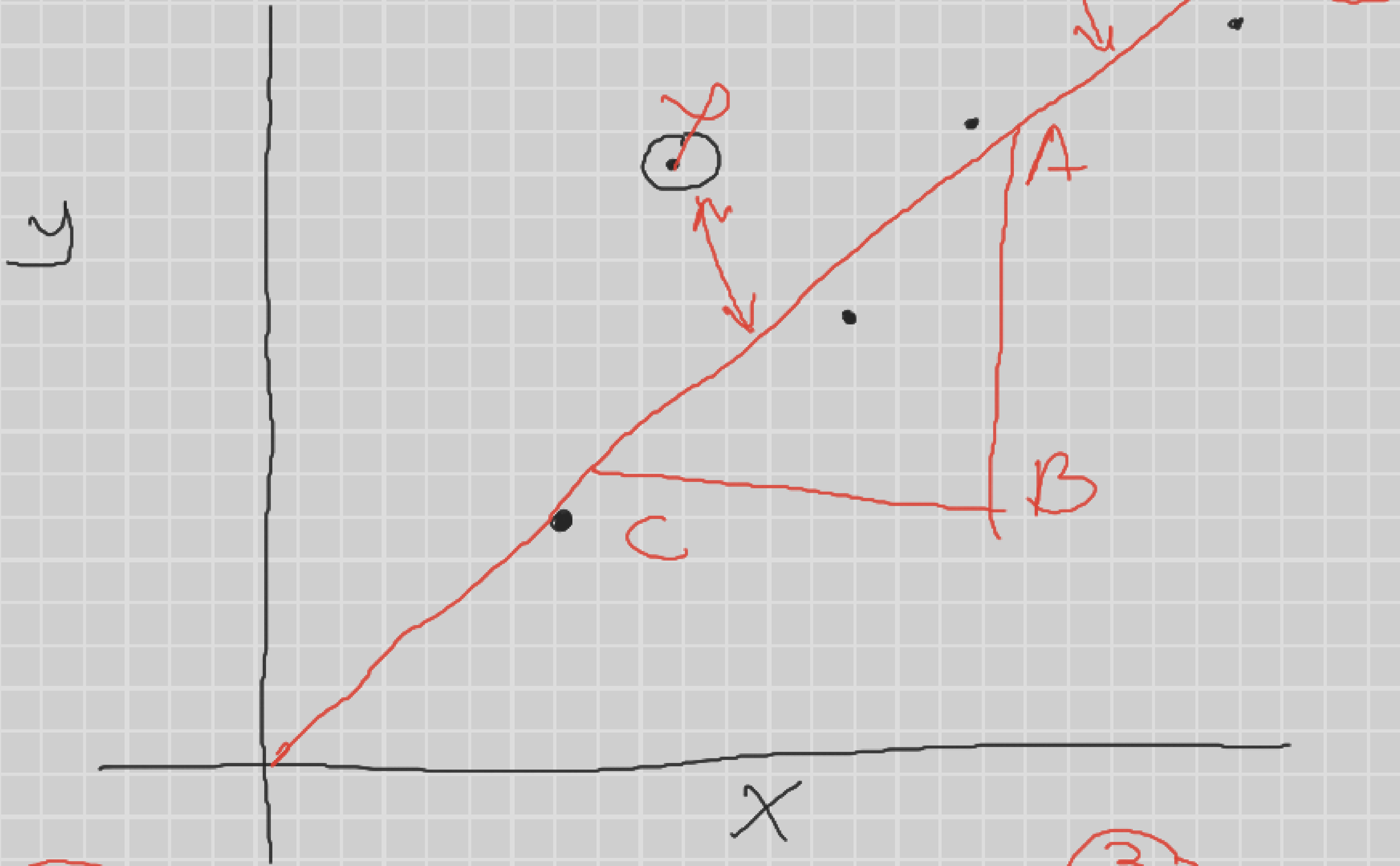
$y = ??$ $y(x) =$

Year	Population (Lakh
1990	20
1995	
2000	40
2010	50
2020	72

$P_n(x) = ??$



curv fitting



$y = mx$

X Y

X Y m

x_1	y_1	$y_1/x_1 = m_1$
x_2	y_2	$y_2/x_2 = m_2$
x_3	y_3	y_3/x_3
$\vdots$		
x_n	y_n	y_n/x_n

$$m = \frac{AB}{BC}$$

3

$$X \left\{ \boxed{m = \frac{m_1 + m_2 + \dots + m_n}{n}} \right.$$

Polynomial

$$f(x) = a_n x^{n} + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$$

① $a_n \neq 0$, n^{th} order poly.

② $(n+1) \rightarrow$ a's value

③ n root of polynomial

$$\gg \quad a = \begin{array}{c} \left[\begin{array}{ccc} 1 & 0 & -1 \end{array} \right] \\ \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ x^2 & x & \end{array} \end{array}$$

$$\underline{\underline{a_2}}x^2 + \underline{\underline{a_1}}\,x + \underline{\underline{a_0}}$$

$$\quad 1 \qquad\quad 0 \qquad -1$$

$$\underline{x^2 - 1 = 0}$$

$x^3 - 2x^2 + 4 = 0$

$X = [1 \quad -2 \quad 0 \quad 4]$

$X = [1 \quad -2 \quad 4]$ — $x^2 - 2x + 4 = 0$

"a" = polyfit (x, y, n)

polval (a, x)

$x^4 - 3x^2 + 4x - 2 = 0$ $y = [\, a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0 \,]$

$a = [1 \;\; 0 \;\; -3 \;\; 4 \;\; -2]$

$a = [1 \;\; -3 \;\; 4 \;\; -2]$ — $x^3 - 3x^2 + 4x - 2 = 0$

1. a = polyfit (x, y, n)

2. polyval

$x \quad y, \; n$

x_0

$\vdots$

x_n

a = polyfit$(x, y, 2)$

$a = [\uparrow \; \uparrow \; \uparrow]$

$x^2 \;\; x$ cont

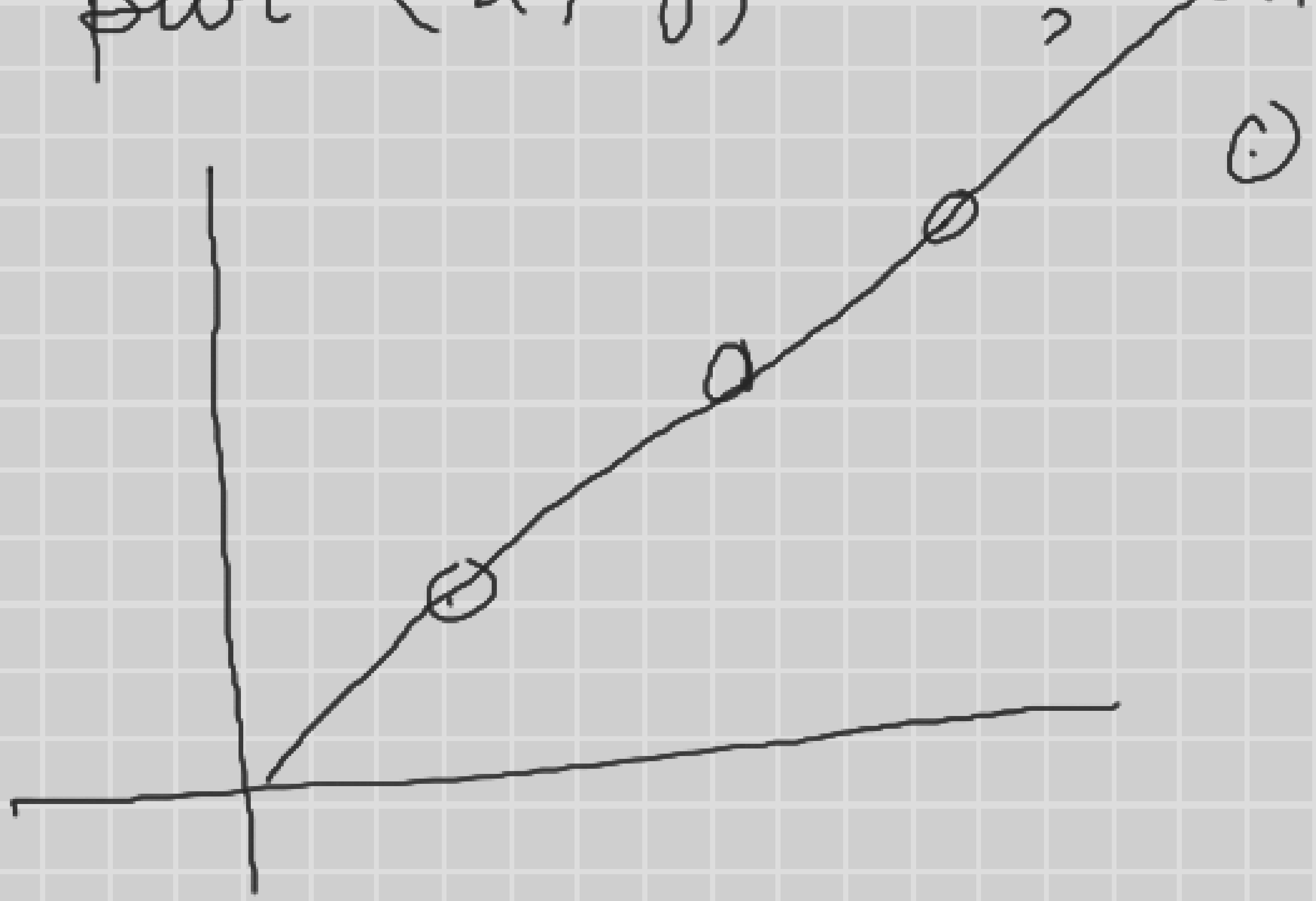
xnew = $(x_0 : x_n)$

ynew = polyval (a, xnew)

x	y
1	1
2	1.8
3	3.2
4	4.1
5	5.0
6	6.1
7.0	7.2

n=1

```
>> x=[                    ];
   y= [                   ];
xnew= 1 : 0.1 : 7.0 ;
a= polyfit (x, y, 1);
ynew= polyval (a, xnew)
plot (x, y, 'o', xnew, ynew)
```



$P(t) = P_0 e^{-t/\tau}$ $P(t)$ = pressure $P_0 = ??$ $\tau = ??$

$$\log P = \log P_0 - t/\tau \quad \text{— (1)}$$

Pbar = log P
tbar = t

n = 1

a = polyfit(tbar, Pbar, 1)

a(1) = -1/tau   tau = -1/a(1);
a(2) = log P0   P0 = exp(a(2))

$$y = x^a x^b$$

```
t = [      ],
P = [      ];
Pbar = log(P);
tbar = t;
a = polyfit(tbar, Pbar, 1);
P0 = exp(a(1));
tau = -1/a(2);
tnew = ______;
Pnew = P0 * e^(-tnew/tau);
plot(t, P, 'o', tnew, Pnew)
```

Roots

1. roots ✓ → valid for polynomial
2. fzero

>> roots(a) ; a=[ ];

$$x^3 - 2x^2 + 5x - 4 = 0$$

✓ a=[1 -2 5 -4];

>> ROOT = roots(a)

root_fzero = fzero ('fun', guess value, accu)

*optional

$10^{-3} - 10^{-7}$

1.0e-3, 1.0e-7

inline

@

$f(x,y) = \underbrace{x^2 + y^2}$

```
f1= inline('x.^2+ y.^2')
f2= @(x,y) x.^2+y.^2
```

Bisection method

$y=f(x)$; $[a, b]$

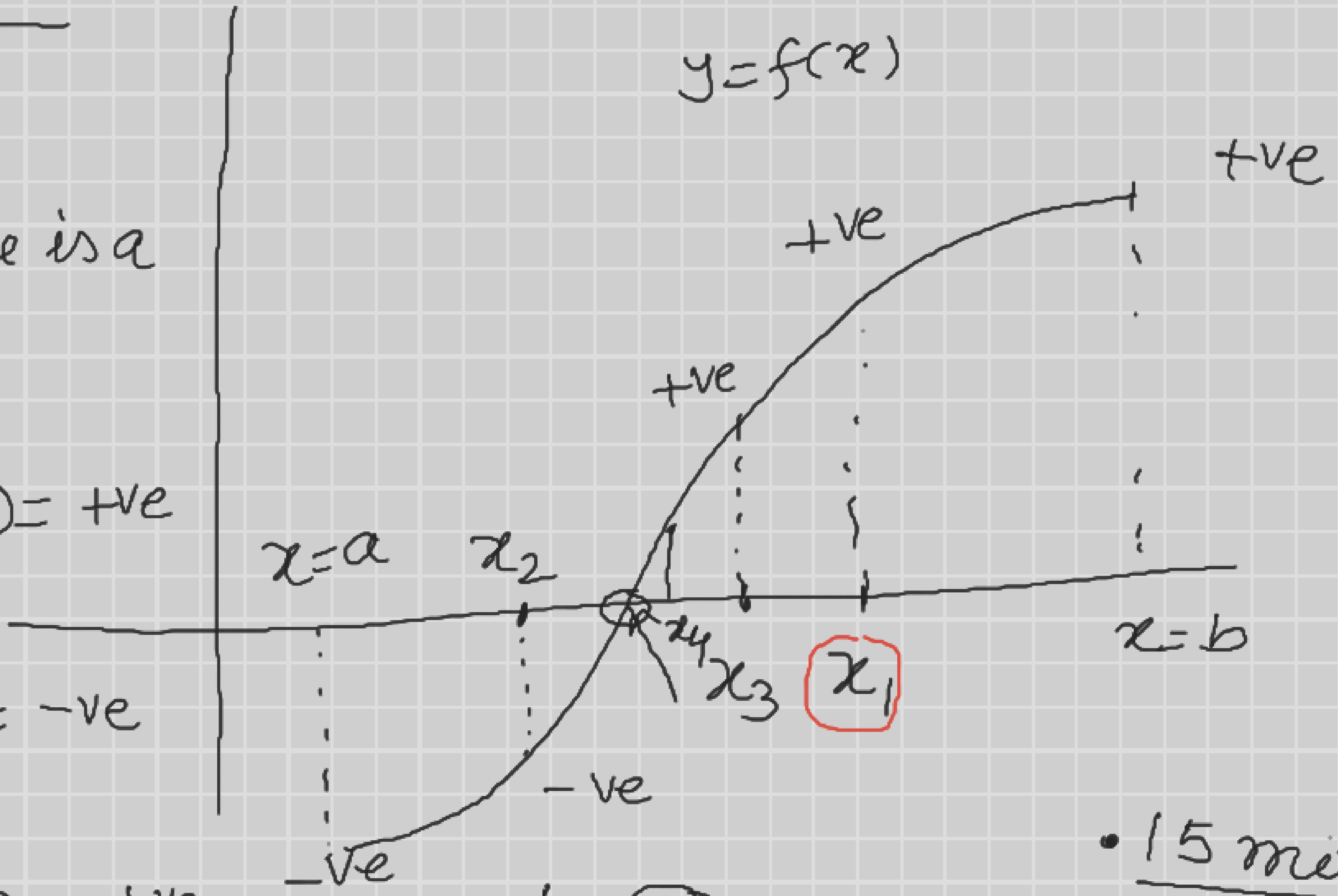
→ $f(a)f(b) < 0$ → there is a root $[a, b]$

→ $x_1 = \frac{(a+b)}{2}$; $f(x_1) = +ve$

→ $x_2 = \frac{(a+x_1)}{2}$; $f(x_2) = -ve$

→ $x_3 = \frac{(x_2+x_1)}{2}$; $f(x_3) = +ve$

→ $x_4 = \frac{(x_2+x_3)}{2}$



1. 3

12

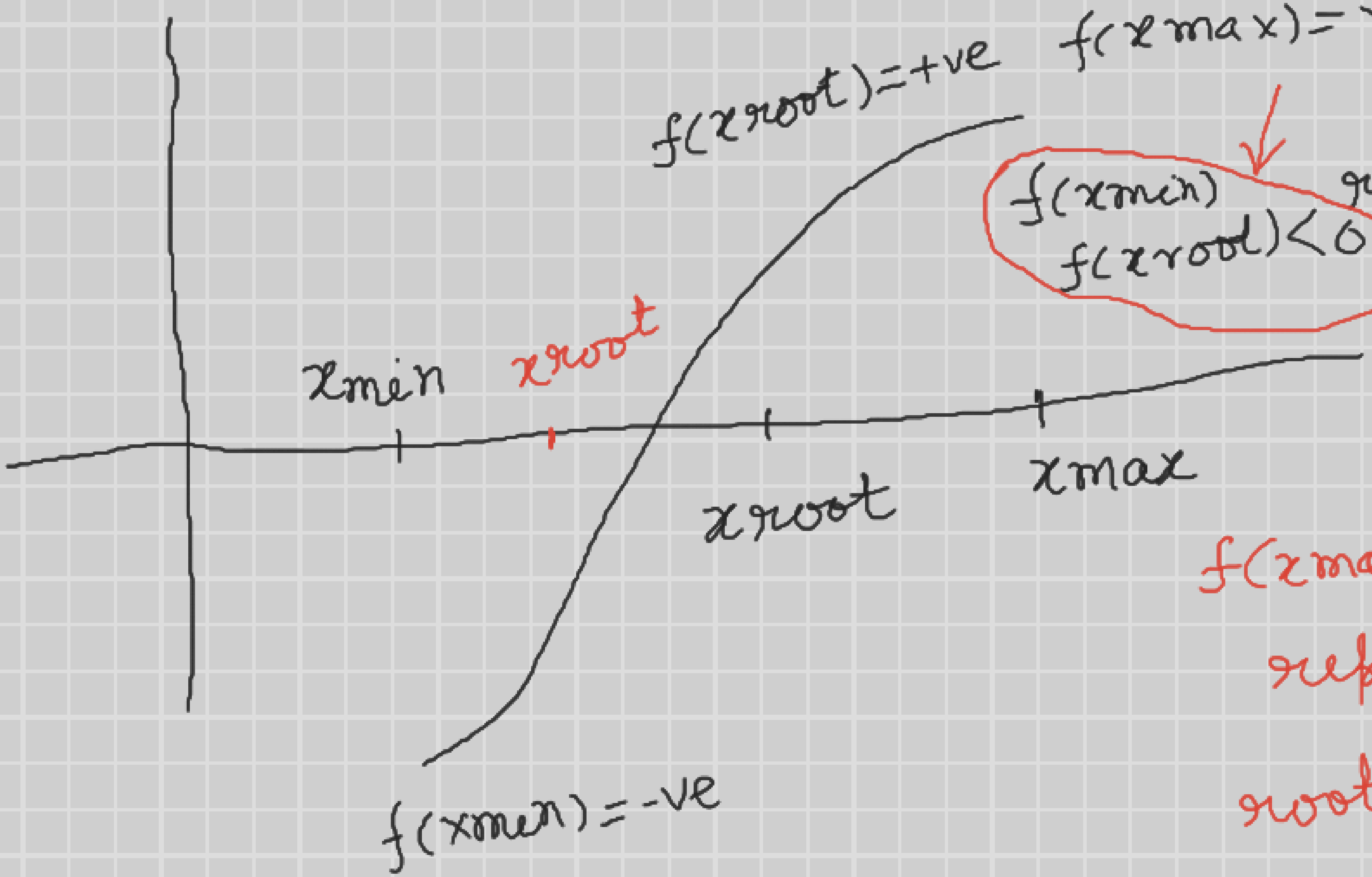
• 15 min

1. 1

2. 2,90

30

90



$$x_{root} = \frac{(x_{min} + x_{max})}{2}$$

$f(x_{min})\, f(x_{root}) < 0 \xrightarrow{\text{replace}}$ x_{max} by x_{root}

$$= \frac{(x_{min} + x_{root})}{2}$$

$f(x_{max})\, f(x_{root}) < 0$

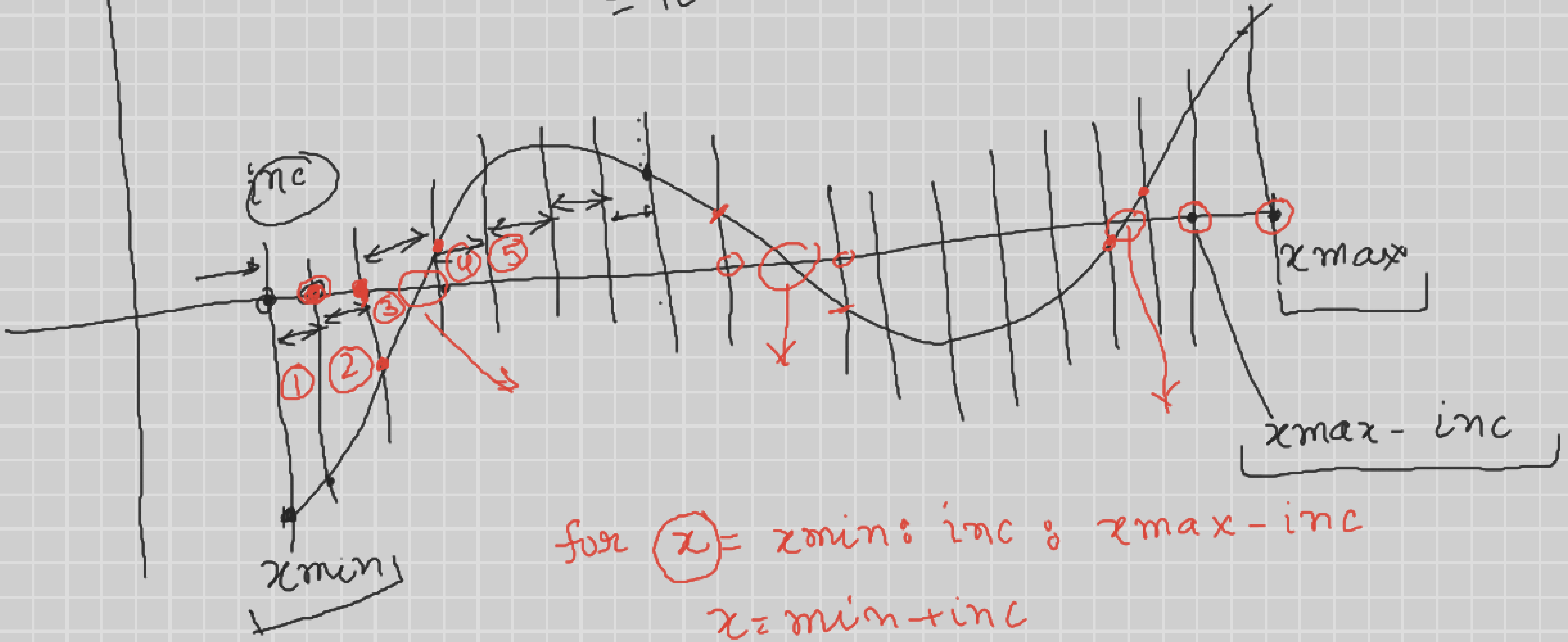
replace x_{min} by x_{root}

$$root = \frac{(x_{root} + x_{max})}{2}$$

nstp

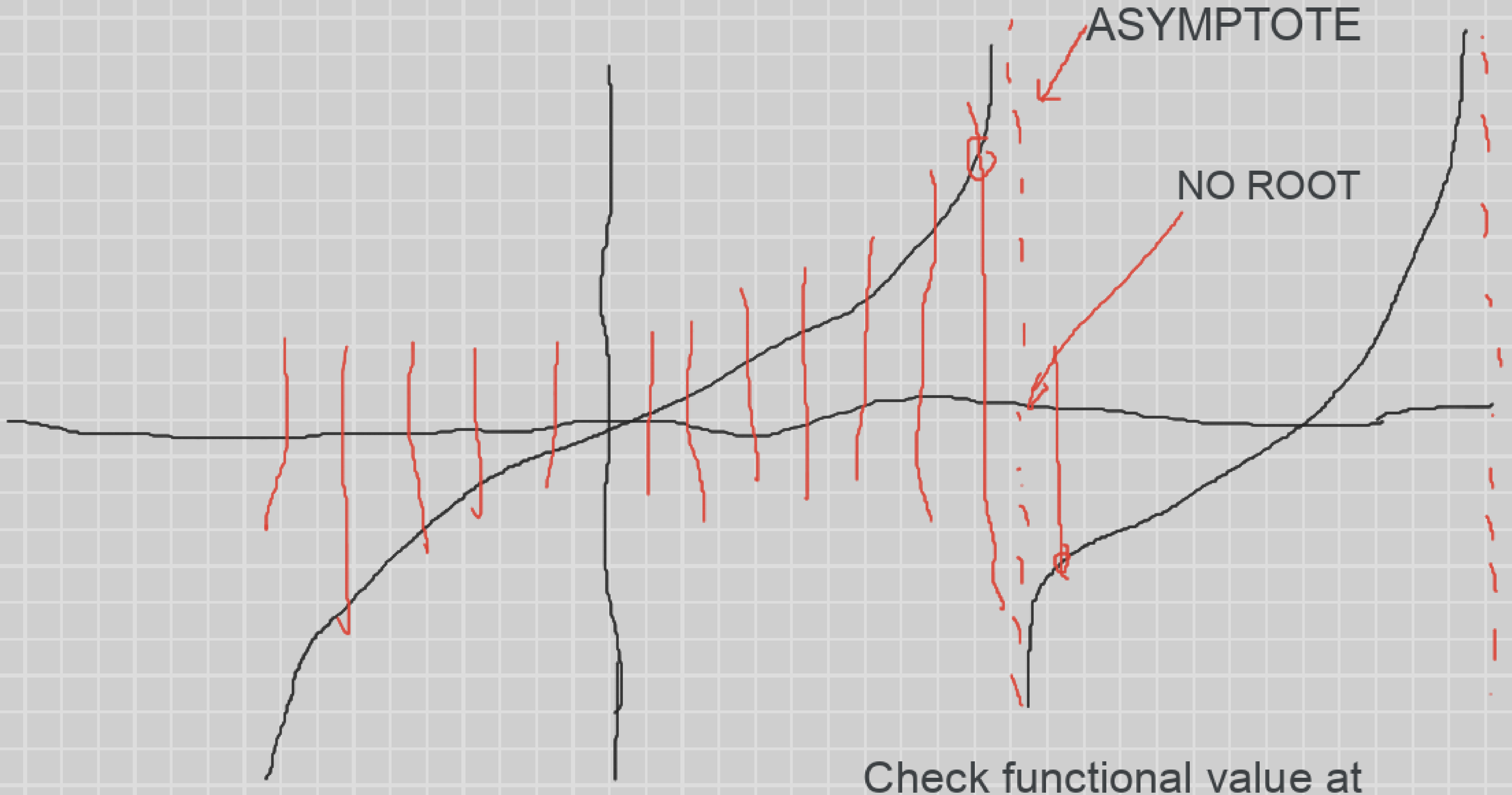
inc = 1.0e-4

$= 10^{-2}$



for x = xmin : inc : xmax - inc

x = min + inc



Check functional value at root..... f(root) should be small

MATLAB CODE FOR INPUT AND OUTPUT

```
%************************************************ **
%     Save the data by using "DISP", "SAVE and fprintf" command
%***************************************************
%   INPUT BY USER: 'input'
%   INPUT FROM FILE: 'load'
x=input('enter x: ')    % numeric input by user
y=input('enter x: ','s') % string input by user

load('input.txt');     %   save data in file input.txt in order to see how data can be read from file
 x=input(:,1);
 y=input(:,2);
 plot(x,y)
```

```
%  OUTPUT
%   1. remove semicolon (;).....
%   2. disp
%   3. save.....   save -ascii filename.exe var1 var2  ( avoid , between var1 and var2)
%   4. fprintf

clear all
a=10;b=40;
 disp(a);
disp('a');      %    print the string a not the numerical value
disp(['the value of a = ',num2str(a), '   &   b = ',num2str(b)])

 x=0:10;
 y=exp(x);
 z=[x' y']
 save -ascii output.txt z


 x=input('enter x: ');
 y=input('enter y: ');
 z=x+y;
fprintf('the addition of the two numbers x = % 5.2f and y = %6.3f is z = % 10.5f\n',x,y,z)

x=0:10;
y=x.^2;
z=[x;y]
fid=fopen('output.dat','w')
fprintf('the output looks like\n')
fprintf('%5.2f %10.1f \n',z)

fprintf(fid,'the output looks like\n');
fprintf(fid,'%5.2f %10.1f \n',z);
fclose(fid)
```

MATLAB CODE FOR INTERPOLATION AND CURVE FITTING

```
%                               INTERPOLATION
% method: linear: data point are connected by line in each interval(polynomial of 1st order)
% method: cubic: data point are connected by polynomial of 3 order in each
% interval
% method: spline: data point are connected by polynomial of lowest order
% piecewise
% data point: (x,y); unknown data point: xi, yi
% syntax: yi=interp1(x,y,xi,'method')
%         yi=spline(x,y,xi)
clear all
x=0:10;
y=sin(x);
xi=0:0.1:10;
yi=interp1(x,y,xi,'linear');
yii=interp1(x,y,xi,'cubic');
yiii=interp1(x,y,xi,'spline');
yiv=spline(x,y,xi);
subplot(2,2,1)
plot(x,y,'o',xi,yi,'r')
legend('DATA POINT','LINEAR FITTED CURVE')
subplot(2,2,2)
plot(x,y,'o',xi,yii,'r')
legend('DATA POINT','CUBIC FITTED CURVE')
subplot(2,2,3)
plot(x,y,'o',xi,yiii,'r')
legend('DATA POINT','SPLINE FITTED CURVE')
subplot(2,2,4)
plot(x,y,'o',xi,yiv,'r')
legend('DATA POINT','SPLINE_2 FITTED CURVE')
```

```
%         ******************
%           CURVE FITTING
%          *****************

% Two steps:
% 1. find out all n+1 a's coefficeint
% y = a(n)*x^n+a(n-1)*x^(n-1)+.......a(0); "polyfit"
% 2. generate the new data point using these a's coefficients: "polyval"

%clear all;
% plot(x,y,'o',xi,yi)
% legend('exp data point','fitted curve')
x=[1 2 3 4 5 6 7];
y=[1.1 2.2 3.0 3.8 5.2 6.2 7.0];
xnew=1:0.1:7;
a=polyfit(x,y,1);
ynew=polyval(a,xnew);
plot(x,y,'o',xnew,ynew,'r')

% Expected function for data P(t)=P(0)exp(-t/tau); find out the coefficient P(0)
t=[0 0.50 1.0 5.0 10.0 20.0];
p=[760 625 528 85 14 0.16];
pbar=log(p);
tbar=t;
a=polyfit(tbar,pbar,1);
tau=-1/(a(1));
p0=exp(a(2));
disp(['coeff: p0 =',num2str(p0), 'and tau =', num2str(tau)]);
tnew=linspace(0,20,100);
pnew=p0.*exp(-tnew/tau);
plot(t,p,'o',tnew,pnew,'r')
pnew=polyval(a,tnew);
plot(t,p,'o',tnew,exp(pnew))
```

MATLAB CODE FOR ROOT

```
%    ******************************************************************
%                          Roots (roots)
%    ******************************************************************
% 1. Polynomial roots search by "roots" inbuilt function
% It gives all REAL and IMAGINARY roots of ploynomial
% a(n)*x^n+a(n-1)*x^(n-1)+........a(1)*x+a(0)=0     POLYNOMIAL FUNCTION
clc
clear all; format long;
A=[1 0 -4];% x^2-4=0; (n+1): a's coefficients in [a(n) a(n-1) a(n-2).....a(0)] order
A1=[1 0 4 2];      % x^3+4x+2=0
Roots_1=roots(A)     % real roots
Roots_2=roots(A1) % gives real and imaginary roots

%    ******************************************************************
%                          Roots (fzero)
%    ******************************************************************

%
%  Defining the function
%  ***************************
f1=inline('sin(x)-exp(x)+5');                        % sin(x)-exp(x)+5 by inline function
f2=@(x)sin(x)-exp(x)+5                               % sin(x)-exp(x)+5 by @ function
%
% ***************************
%  Syntax for "fzero" command
% ***************************
x_sol_1=fzero(f1,1)
x_sol_2=fzero(f2,1)
```

```
%  ******************************************************************
% BISECTION METHOD (ONLY ONE ROOT)
%  ******************************************************************
clear all
xmin=-1.9;
xmax=-2.3;
nstp=10;
f=@(x) sin(x)-exp(x)+5;
ROOT=[];
if f(xmin)*f(xmax)>0
 disp('there is no root in given interval')
else
  for i=1:nstp
  xroot=(xmin+xmax)/2;
  ROOT=[ROOT; xroot];
  if f(xmax)*f(xroot)>0
     xmax=xroot;
   else
     xmin=xroot;
   end
  end
end

disp(ROOT)

% ******************************************************************
%  ALL ROOTS AVOIDING ASYMPTOTE USING BISECTION METHOD
% ******************************************************************
clear all
xmin=-2*pi-0.01;
xmax=2*pi+0.01;
inc=1.0e-4;
##f=@(x) sin(x)-exp(x)+5;
f=@(x) tan(x)
ROOT=[];
for x=xmin:inc:xmax-inc
  if f(x)*f(x+inc)<0 && f((x+x+inc)/2)<1.0e-2
   xroot=(x+x+inc)/2;
   ROOT=[ROOT;xroot];
  else
  end
end
disp(ROOT)
```