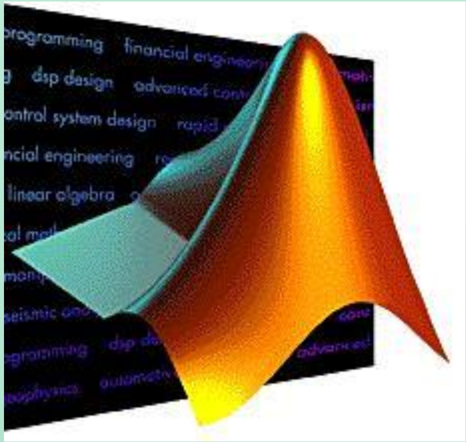
NONLINIER EQUATIONS



- Many problems in engineering and science require the solution of nonlinear equations. Several examples of such problems drawn from the field of chemical engineering and from other application areas are discussed in this section.
- The methods of solution are developed in the remaining sections of the chapter, and specific examples of the solutions are demonstrated using the MATLAB software.

- A **system of nonlinear equations** is a system of two or more equations in two or more variables containing at least one equation that is not linear.
- Example:

$$x - y = -1 \ \text{(linier)}$$

$$y = x^2 + 1 \ \text{(non linier)}$$

$x - y = -1$ (1)

$y = x^2 + 1$ (2)

<u>Penyelesaian analitis</u>:

dengan substitusi dari pers (1)

$y = x + 1$ ke pers (2)

$x + 1 = x^2 + 1$

$x^2 - x = 0$ atau $x(x - 1) = 0$

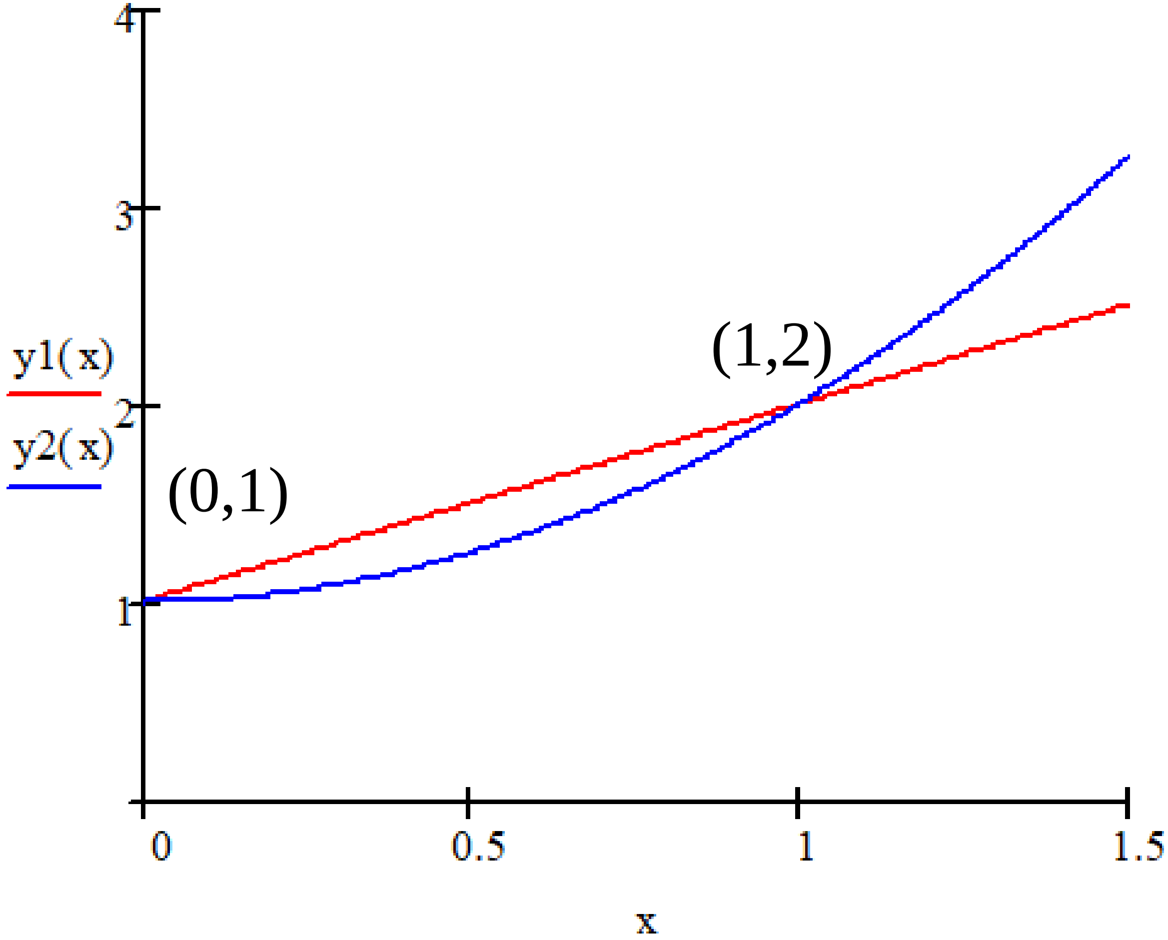
$x = 0$ maka $y = 1$

$x = 1$ maka $y = 2$

04 Nonlinier Equations

$$y1(x) := x + 1$$

$$y2(x) := x^2 + 1$$



- The solution of nonlinier equations to find x for a function x, f(x) = 0.

- Example, $f(x) = e^{-x} - x$,

  how much x for f(x) = 0.

Solution of nonlinier equations

The two major classes of methods available are distinguished by the type of initial guess.

They are

- *Bracketing methods*. As the name implies, these are based on two initial guesses that bracket the root.
- *Open methods*. These methods can involve one or more initial guesses, but there is no need for them to bracket the root.

We study :

- Graphical method
- Bisection Method
- Secant Method
- Newton – Raphson Method
- Using MATLAB built in function: fzero, fsolve

Graphical method

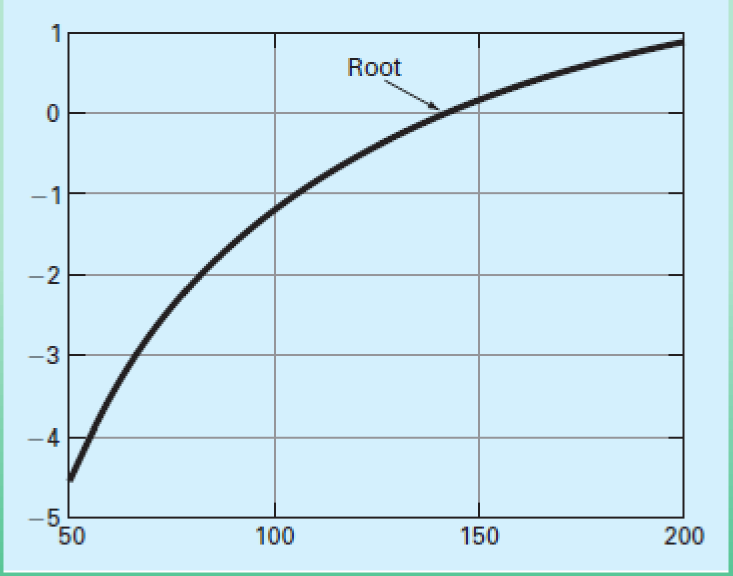
- A simple method for obtaining an estimate of the root of the equation $f(x) = 0$ is to make a plot of the function and observe where it crosses the x axis.
- This point, which represents the x value for which $f(x) = 0$, provides a rough approximation of the root.

- Use the graphical approach to determine the mass of the bungee jumper with a drag coefficient of 0.25 kg/m to have a velocity of 36 m/s after 4 s of free fall. Note: The acceleration of gravity is 9.81 m/s^2.

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}t\right)$$

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}t\right) - v(t)$$

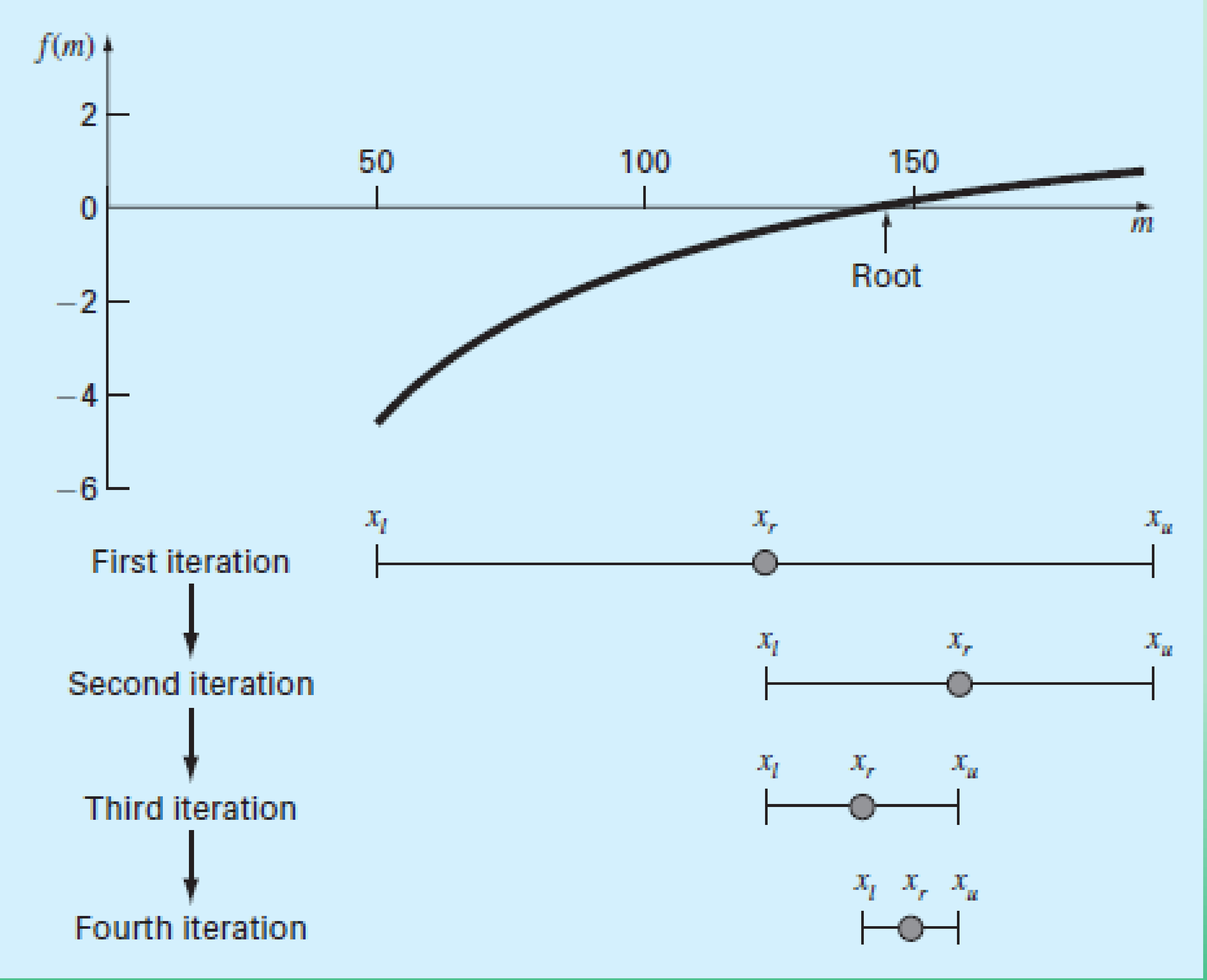
```
>> cd = 0.25; g = 9.81; v = 36; t = 4;
>> mp = linspace(50,200);
>> fp = sqrt(g*mp/cd).*tanh(sqrt(g*cd./mp)*t)-v;
>> plot(mp,fp),grid
```



Bisection Method

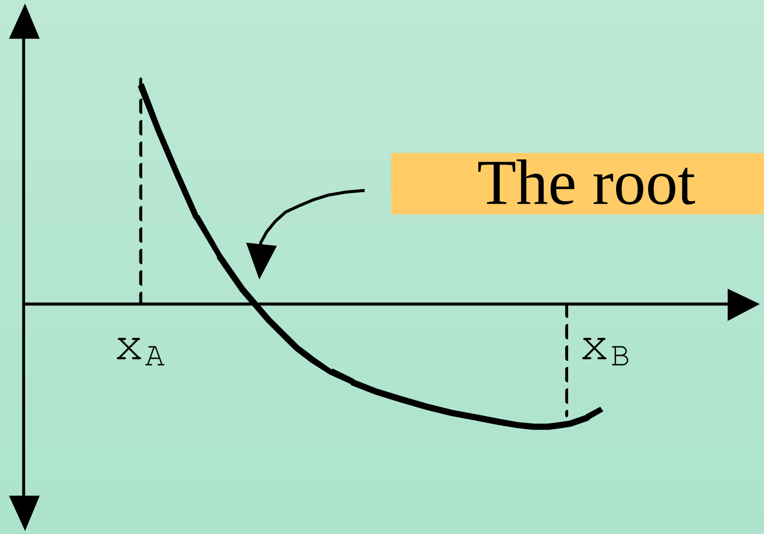
- The *bisection method* is a variation of the incremental search method in which the interval is always divided in half.
- If a function changes sign over an interval, the function value at the midpoint is evaluated.
- The location of the root is then determined as lying within the subinterval where the sign change occurs.
- The subinterval then becomes the interval for the next iteration. The process is repeated until the root is known to the required precision.

- A graphical depiction of the method is provided in this Fig.



Step by step for solution

1. Determine initial value x_A dan x_B
2. Find $f(x_A)$ dan $f(x_B)$
3. The interval is correct if $f(x_A)$ & $f(x_B)$ different sign



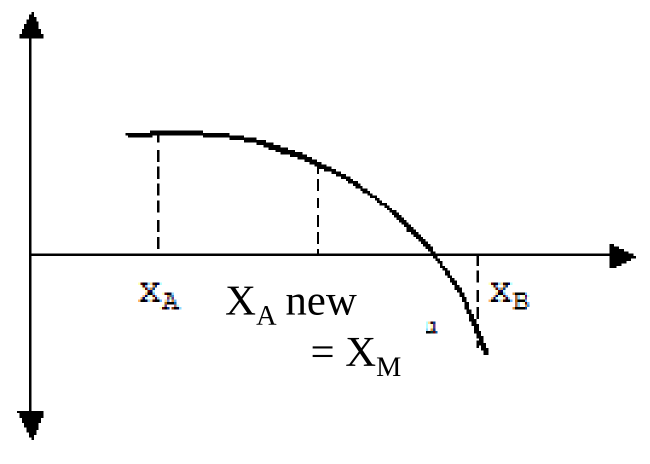
check $[f(x_A)]x[f(x_B)] < 0$.

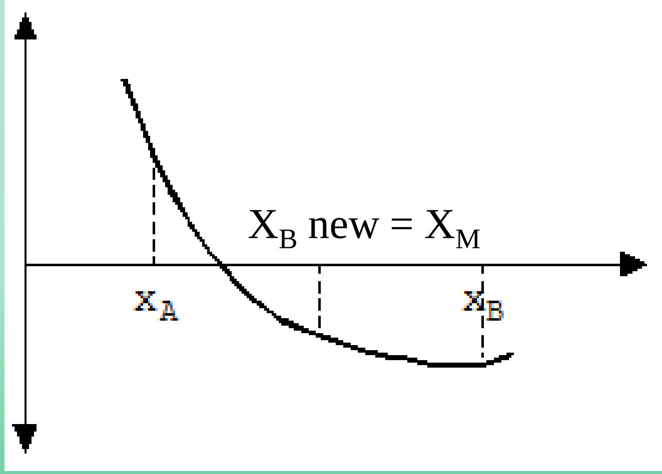
If not, repeat (1) & (2).

4. Find x_M (middle point)

$$x_M = \frac{x_A + x_B}{2}$$

5. Find $f(x_M)$
   a) If $[f(x_A)]x[f(x_M)] > 0$, then x_A new = x_M & $x_B = x_B$
   b) If $[f(x_A)]x[f(x_M)] < 0$, then $x_A = x_A$ & x_B new = x_M





6. Check $f(x_m) <$ Tolerance. If not, repeat (4) for new x_m .

Contoh 3.1. Konversi untuk disosiasi H_2O

Uap air didisosiasikan (dipecah) menjadi H_2 dan O_2 pada tekanan 0,2 atm sebagai berikut:

$$H_2O \leftrightarrow H_2 + \tfrac{1}{2} O_2$$

Fraksi molekul (x) dari H_2O dapat dinyatakan sebagai berikut:

$$k_p = \frac{x}{(1-x)}\sqrt{\frac{2p}{(2+x)}}$$

Jika $k_p = 0{,}4568$ tentukanlah x yang memenuhi persamaan di atas.

```
clc
clear all

disp 'Masukkan Perkiraan Batas Bawah dan Batas Atas '
xa=input('batas bawah = ');
xb=input('batas atas  = ');
fa = contoh31(xa);
fb = contoh31(xb);

x=[xa:(xb-xa)/50:xb];
plot(x,contoh31(x));
xlabel('x')
ylabel('f(x)')
grid on

while fa*fb > 0
   disp 'Tebakan salah, ganti batas atas dan/atau batas bawah'
   disp  'Perhatikan  grafik!!  Perkirakan  nilai  x  untuk  f(x)
mendekati nol'
   xa=input('batas bawah = ');
   xb=input('batas atas  = ');
   fa=contoh31(xa);
   fb=contoh31(xb);
end
tol=0.000001;

xm=(xa+xb)/2;
fm=contoh31(xm);
while abs(xa-xb)>tol
   xm=(xa+xb)/2;
   fm=contoh31(xm);
   if fa*fm<0
      xb=xm;
      fb=contoh31(xb);
   else
      xa=xm;
      fa=contoh31(xa);
   end
end
t =['f(x)=0 untuk x = 'num2str(xm)];
disp (t)
```

```
function f_x = contoh31(x)
p = 0.2;
kp = 0.4568;
f_x = x./(1-x).*sqrt(2*p./(2+x))-kp;
```

Contoh 3.2. Faktor Kompresibilitas Gas Ideal

Hubungan faktor kompresibilitas gas ideal dalam bentuk

$$z = \frac{1 - y + y^2 - y^3}{1 - y^3}$$

dengan y = b/4v, untuk b adalah koreksi van der Waals dan v adalah volum molar. Jika z = 0,892 berapakah y ?

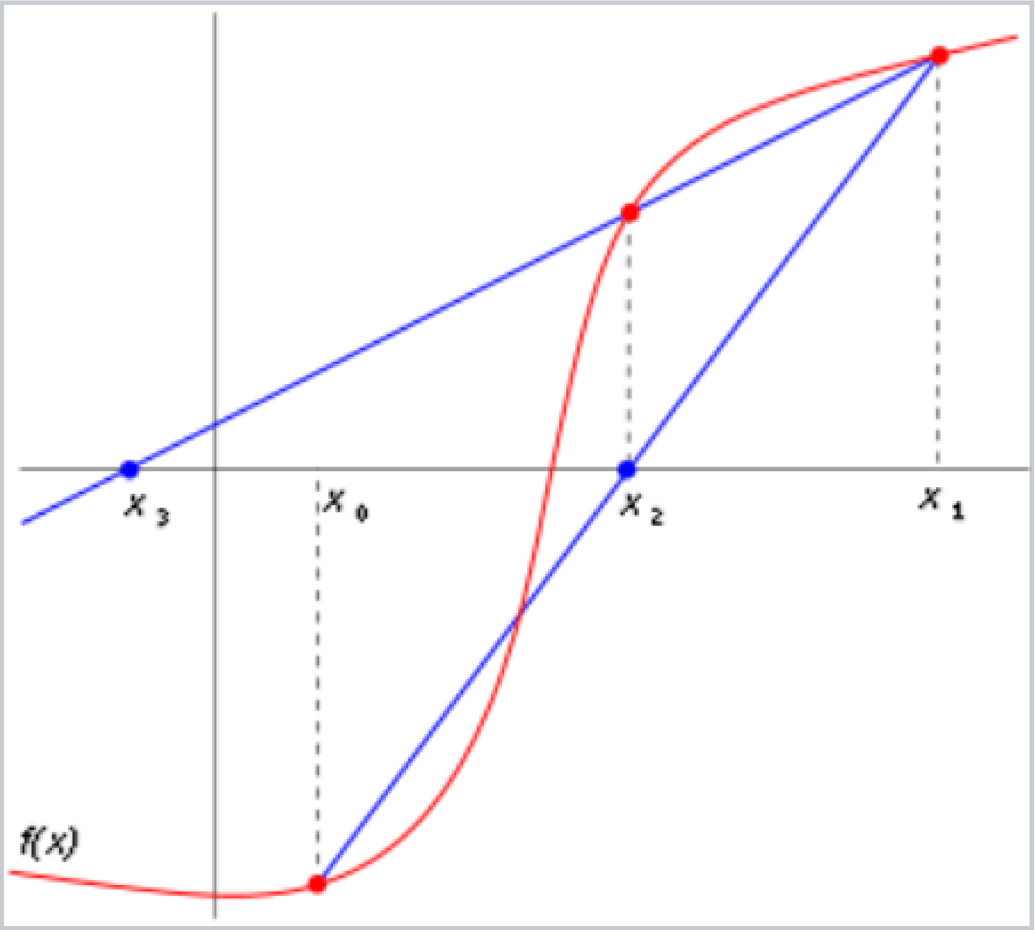
Penyelesaian dilakukan dengan program bisection yaitu dengan mengganti fungsi disosiasi H_2O dengan fungsi untuk persoalan faktor kompresibilitas gas ideal.

```
function f_y = contoh32(y)
%   Faktor Kompresibilitas Gas Ideal
%                1 - y + y^2 - y^3
%           z = -------------------
%                    1 - y^3
%            z = 0,896

%          Nama File : contoh32.m
%          Surakarta, Oktober 2005
%   ---------------------------------------

z = 0.892;
f_y = (1-y+y.^2-y.^3)-z.*(1-y.^3);
```

Secant Method



The first two iterations of the secant method. The red curve shows the function *f*, and the blue lines are the secants. For this particular case, the secant method will not converge to the visible root.

Starting with initial values x_0 and x_1, we construct a line through the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$, as shown in the picture above. In slope–intercept form, the equation of this line is

$$y = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1) + f(x_1).$$

The root of this linear function, that is the value of x such that $y = 0$ is

$$x = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}.$$

We then use this new value of x as x_2 and repeat the process, using x_1 and x_2 instead of x_0 and x_1. We continue this process, solving for x_3, x_4, etc., until we reach a sufficiently high level of precision (a sufficiently small difference between x_n and x_{n-1}):

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)},$$

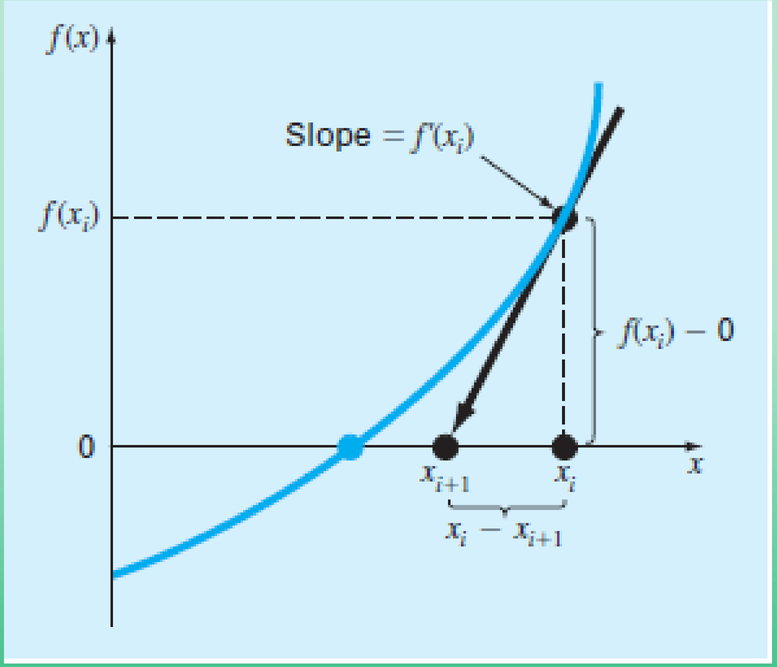
$$x_3 = x_2 - f(x_2)\frac{x_2 - x_1}{f(x_2) - f(x_1)},$$

$$\vdots$$

$$x_n = x_{n-1} - f(x_{n-1})\frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}.$$

Newton-Rahpson Method

- Newton-Raphson Method is one of the most powerful and well-known numerical methods for solving a root-finding problem.
- If the initial guess at the root is *xi* , a tangent can be extended from the point [*xi* , *f (xi)*]. The point where this tangent crosses the *x* axis usually represents an improved estimate of the root.



Slope for x_i is derivatif function f(x)

$$f'(x) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Step by step solution

1. Determine the initial guess x_i
2. Find $f(x_i)$ & $f'(x_i)$
3. Calculate x_{i+1} with equation
4. Check $| x_{i+1} - x_i | <$ tolerance atau

   $| f(x_i) | <$ tolerance,

   If yes, then finished.

   If not, x_i new= x_{i+1}, repeat form step (2)

If f'(x) is difficult to be obtained by analytical method, we can use numerical methods.

Forward difference

$$f'(x) \cong \frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

Backward difference

$$f'(x) \cong \frac{f(x) - f(x-\varepsilon)}{\varepsilon}$$

Central difference

$$f'(x) \cong \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\,\varepsilon}$$

with ε is very small value.

Problem Statement. Use the Newton-Raphson method to estimate the root of $f(x) = e^{-x} - x$ employing an initial guess of $x_0 = 0$.

Solution. The first derivative of the function can be evaluated as

$$f'(x) = -e^{-x} - 1$$

which can be substituted along with the original function into Eq. (6.6) to give

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

i	x_i	ε_t (%)
0	0	100
1	0.500000000	11.8
2	0.566311003	0.147
3	0.567143165	0.0000220
4	0.567143290	$< 10^{-8}$

Contoh 3.3. Temperatur Dew Point untuk Campuran Benzena dan Toluena

Tentukan temperatur dew point (Titik Embun) dan komposisi liquid dari suatu campuran gas benzena dan toluena pada tekanan 1 atm (760 mmHg). Komposisi uap adalah 0,77 fraksi mol benzena dan 0,23 fraksi mol toluena.

Campuran gas dan liquid diasumsikan sebagai campuran ideal. Kondisi kesetimbangan sesuai dengan Hukum Roult-Dalton, $y_iP=x_iP_i^O$ Tekanan uap murni dihitung dengan persamaan $\log p^O = A - \dfrac{B}{T+C}$ untuk p^O dalam mmHg dan T dalam ^{O}C

	Benzena	Toluena
A	6,89745	6,95334
B	1206,35	1343,94
C	220,237	219,377

```
clc
clear all
xnew=input(' Nilai untuk tebakan awal =  ');
xold=0;
tol=0.0001;
eps=0.0001;

disp '   x-old        f(x-old) '
while abs(xnew-xold)>tol
    xold=xnew;
    fxold=contoh33(xold);
    t=['   ' num2str(xold) '      ' num2str(fxold) ];
    disp (t)
    fmin=contoh33(xold-eps);
    fplus=contoh33(xold+eps);
    dfx=(fplus-fmin)/2/eps;
    xnew=xold-fxold/dfx;
end
t=['akar persamaan, x = ' num2str(xold) '   f(x) = ' num2str(fxold)];
disp(t)
```

```
function fT=contoh33(T)
P=760;            % Konversi dari atm ke mmHg
y1=0.77;
y2=0.23;
p1o=10^(6.89745-1206.35/(T+220.237));
p2o=10^(6.95334-1343.94/(T+219.377));
x1=y1*P/p1o;
x2=y2*P/p2o;
fT=x1+x2-1;
```

Contoh 3.4. Hubungan Faktor Friksi suatu Pelarut dengan Bilangan Reynolds

Hubungan faktor friksi untuk aliran suatu pelarut dengan bilangan Reynolds (Re) secara empiris adalah

$$\frac{1}{\sqrt{f}} = \left(\frac{1}{k}\right)\ln\left(\mathrm{Re}\,\sqrt{f}\right) + \left(14 - \frac{5{,}6}{k}\right)$$

dengan k = konsentrasi larutan dan f adalah faktor friksi. Tentukanlah f, jika Re = 3750 dan k=0,28

Program penyelesaian dengan Metode Newton-Raphson. Karena bekerja pada bilangan yang lebih kecil, toleransi dapat kita turunkan sampai 10^{-7}.

```
function f_f = contoh34(f)
Re = 3750;
k = 0.28;
f_f = ((1/k)*log(Re*sqrt(f))+(14-5.6/k))*sqrt(f)-1;
```

Penyelesaian dengan MATLAB

- Mencari nilai suatu variabel dalam satu persamaan → gunakan **fzero**
- Mencari nilai multi variabel dalam lebih dari satu persamaan → gunakan **fsolve**

fzero function

- The fzero function is designed to find the real root of a single equation. A simple representation of its syntax is :

x = fzero('*function*',xo)

where *function* is the name of the function being evaluated, and xo is the initial guess.

fsolve function

Untuk sistem persamaan non linier multi variabel dapat digunakan ‘**function fsolve**’:

This example shows how to solve two nonlinear equations in two variables. The equations are

$$\begin{aligned} e^{-e^{-(x_1+x_2)}} &= x_2\left(1+x_1^2\right) \\ x_1\cos\left(x_2\right)+x_2\sin\left(x_1\right) &= \tfrac{1}{2}. \end{aligned}$$

Convert the equations to the form $\boldsymbol{F}(x)=\mathbf{0}$.

$$\begin{aligned} e^{-e^{-(x_1+x_2)}} - x_2\left(1+x_1^2\right) &= 0 \\ x_1\cos\left(x_2\right)+x_2\sin\left(x_1\right) - \tfrac{1}{2} &= 0. \end{aligned}$$

```
function F = root2d(x)
F(1) = exp(-exp(-(x(1)+x(2)))) - x(2)*(1+x(1)^2);
F(2) = x(1)*cos(x(2)) + x(2)*sin(x(1)) - 0.5;
end
```

Program bisa ditulis di command window atau dalam script tersendiri:

```
clear
clc
fun = @root2d;
x0 = [0,0];
x = fsolve(fun,x0)
```

Hasil run:

```
Equation solved.

fsolve completed because the vector of function
values is near zero
as measured by the default value of the function
tolerance, and
the problem appears regular as measured by the
gradient.

<stopping criteria details>


x =

    0.3532    0.6061
```

- Contoh lain:

$$2 \cdot x + y = 5 - 2 \cdot z^2$$

$$y^3 + 4 \cdot z = 4$$

$$x \cdot y + z = e^z$$

Tentukan nilai x, y, z

Function Fun1 dibuat dalam script

```
function F = Fun1(x)
    F(1)=2*x(1)+x(2)-5+2*x(3)^2;
    F(2)=x(2)^3+4*x(3)-4;
    F(3)=x(1)*x(2)+x(3)-exp(x(3));
end
```

Program dibuat dalam script, sbb:

```
clear
clc
fun = @Fun1;
x0 = [0,0,0];
x = fsolve(fun,x0);
fprintf('x = %2.5f  \n',x(1));
fprintf('y = %2.5f  \n',x(2));
fprintf('z = %2.5f  \n',x(3));
```

Hasil run:

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>

x = 1.42247
y = 0.97539
z = 0.76801
```