KOMPUTASI FISIKA PART 2

MATERI

- Akar-akar Persamaan

  Metode Akolade
  - Metode Grafik
  - Metode Bagi Dua
  - Metode Posisi Salah

  Metode Terbuka
  - Iterasi Satu Titik Sederhana
  - Metode Newton-Raphson
  - Metode Secant
  - Akar Ganda

- Sistem Persamaan Aljabar Linier
  - Eliminasi Gauss
  - Gauss-Jordan
  - Matriks Invers
  - Gauss-Seidel

Akar-akar Persamaan

Tujuan :

- To understand why we as Engineers are interested in finding the roots of algebraic equations
- •To understand the difference between *Open* (metode terbuka *)* and *Bracketing methods (* Metode Akolade *) & to know why* and when we would choose to apply one method over another
- To be able to implement your own root finding tools

Dasar

- Persamaan Matematika f(x) = 0

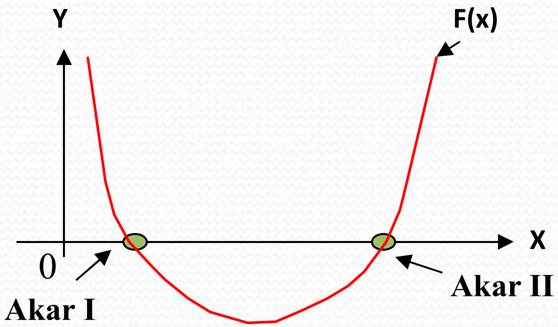
Polinomial atau suku banyak secara umum dinyatakan sbb :

$$a_o + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_N x^N = 0 \ldots\ldots\ldots(1)$$

dengan $\boldsymbol{a_i}$ adalah kontanta.

Akar dari sebuah persamaan adalah harga x yang membuat nilai f(x) = 0.

Contoh kasus :



Pada x_1 nilai $f(x) = y_1 = 0$

Pada x_2 nilai $f(x) = y_2 = 0$

** Kesimpulan : ***x_1 dan x_2 adalah akar - akar dari persamaan $f(x)$***

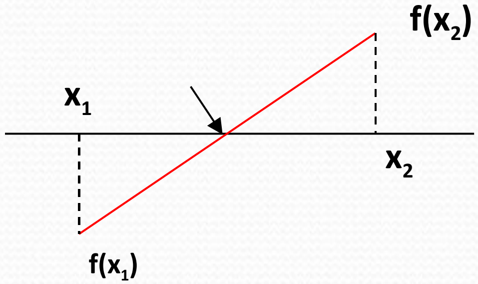
Note :

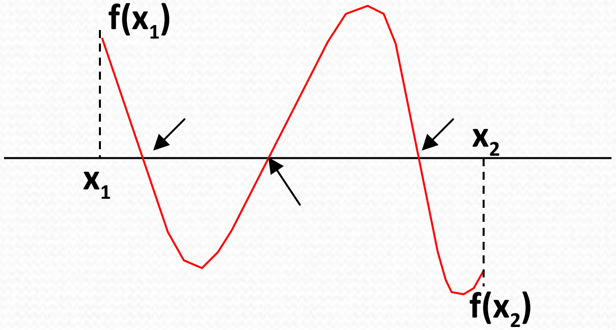
- Khusus untuk polinomial berderajat dua , sebagai alternatif bahwa pencarian akar polinomial dapat dicari dengan rumus kuadratik
- Jumlah akar dari suatu polinomial adalah sama dengan **derajat dari polinomial.**

Ide dasar pencarian akar:

1) jika $f(x1)$ dan $f(x2)$ berlawanan tanda maka

   $f(x1) \times f(x2) < 0$ berarti interval antara x1 dengan x2 terdapat jumlah ganjil dari akar



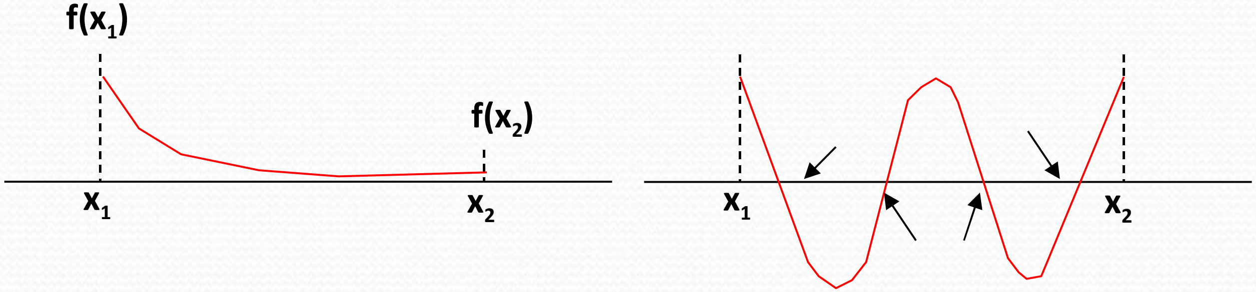


Contoh (a) dan (b) memenuhi $f(x1) \times f(x2) < 0$

Pada (a) ada 1 akar antara interval x1 dan x2

Pada (b) ada 3 akar antara interval x1 dan x2

2) Jika $f(x1)$ dan $f(x2)$ sama tanda maka $f(x1) \times f(x2) > 0$ berarti interval antara x1 dan x2 terdapat jumlah genap dari akar atau tidak ada akar.



Contoh (a) dan (b) memenuhi $f(x1) \times f(x2) > 0$

Pada (a) tidak ada akar antara interval x1 dan x2

Pada (b) terdapat 4 akar antara interval x1 dan x2

3). Jika $f(x_1) \times f(x_2) = 0$ berarti x_1 atau x_2 merupakan akar.

Konsep di atas menjadi **ide** bagi metode numerik Akolade untuk memeriksa apakah dalam interval yang akan diperiksa terdapat akar atau tidak.

Kenapa pake metode numerik

- **Persamaan eksplisit**

  $y - mx + b = 0$

  the unknown is specified explicitly in terms of the other variables & parameters

- **Persamaan implisit**

$$v - \frac{gm}{c}\left(1 - e^{-\frac{c}{mt}}\right) = 0$$

The desired variable can not be isolated on the LHS of an equation

➔Numerical methods must be applied to obtain roots of implicit equations

Graphical Method

- Make a plot of *f(x)*
- Observe where it crosses the x-axis
- Plug the estimate back into the original equation to check
  – "Rough Estimate" approach
  – Limited due to imprecision
  – Provides some understanding of the functions behavior
  – **May be hard to find all roots due to varying ranges.**

  Jarang dipakai…

Bisection Method

Utilize the notion that the function must change signs on either side of the root.

Metode Bisection (metode bagi dua) membagi interval (antara x_1 dan x_2) dimana diperkirakan terdapat **sebuah akar**, menjadi 2 subinterval yang sama besar.

Akar tsb dicari dalam salah satu subinterval dan Interval tidak boleh terlalu lebar. Mengapa ? (PR ya..)

Algoritma metode Bisection :

Langkah 1 : Pilih taksiran nilai x_l sebagai batas bawah interval dan taksiran nilai x_u sebagai batas atas interval.

Jika terpenuhi kondisi :

- $f(x_l) \times f(x_u) < 0$ maka ada akar dalam interval. Selanjutnya ke langkah 2.
- $f(x_l) \times f(x_u) > 0$ maka tidak ada akar dalam interval. Geser posisi interval.
- $f(x_l) \times f(x_u) = 0$ maka x_l atau x_u , salah satu merupakan akar.

- **Langkah 2 :** Taksiran akar yang pertama x’ dimana

$$x' = \frac{x_l + x_u}{2}$$

Langkah 3: Evaluasi keberadaan akar, apakah dalam subinterval pertama (antara xl dan x') atau dalam subinterval kedua (antara x' dan xu).

Jika diperoleh

a) $f(xl) \times f(x') < 0$: akar berada dalam subinterval pertama maka xu = x'. Selanjutnya ke langkah 4.

b) $f(xl) \times f(x') > 0$: akar berada dalam subinterval kedua maka xl = x'. Selanjutnya ke langkah 4.

c) $f(xl) \times f(x') = 0$: x' adalah akar

Langkah 4 : Kembali ke langkah 2 dan proses hingga langkah 3.

BUATLAH FLOWCARTNYA...

Ketentuan program komputasi

- Kriteria terminasi dalam metode bisection menggunakan pendekatan kesalahan aproksimasi, yang ditentukan dengan memakai penaksiran kesalahan:

$$\varepsilon_a = \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| \times 100\,\%$$

- Pada akhir komputasi, akar yang diperoleh sebaiknya dimasukkan kembali dalam fungsi untuk menghitung apakah hasilnya mendekati nol. Pengecekan ini penting untuk mengantisipasi kejadian konvergensi berosilasi atau kejadian yang menyebabkan harga kecil padahal penyelesaian masih jauh dari akar.
- Program harus mempunyai input batasan jumlah iterasi sehingga pada kejadian osilasi, konvergensi perlahan atau penyelesaian divergensi program tetap dapat berhenti

Ex. Program dlm MATLAB

```
iter = 0                              %initialize iteration counter
While ea>es and iter <= imax
xrold = xr
xr = (xl + xu)/2                      % Estimate new root
iter = iter + 1                       % increment iteration counter
if xr ≠ 0 then
ea = abs((xr – xrold)/xr)*100         % calculate approx. rel. error
endif
test = f(xl)*f(xr)                    % good spot to use a Matlab function
if test < 0 then
xu = xr
else if test > 0 then
xl = xr
else
ea = 0
endif
end while loop
Bisect = xr
```

TUGAS 1 PRAKTIKUM MINGGU DEPAN

- buatlah rancangan dari program pencarian akar dengan metode Bisection (di rumah).
- Ketikan rancangan program anda dalam komputer hingga program dapat dieksekusi dengan benar (di laboratorium)
- Cobalah beri beberapa masukan untuk nilai x_1 tetap sementara nilai step bervariasi. Begitu pula coba untuk nilai x_1 bervariasi sementara nilai step tetap. Bandingkan kedua macam perlakuan tsb dan berilah komentar tentang nilai step dan nilai x_1 dihubungkan dengan hasil kerja program dari hasil pengamatan anda.
- Cobalah beri masukan sebagai berikut dan laporkan hasilnya

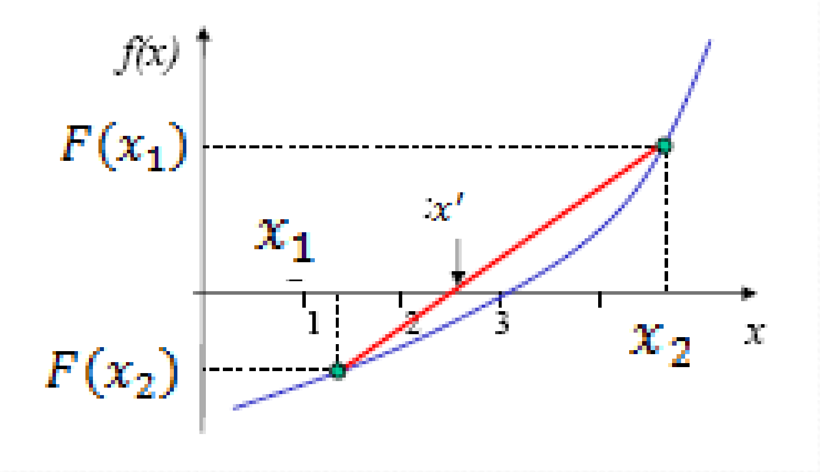
  (persamaannya liat di blog saya hari Rabu)Carilah akarnya
- Cetaklah paparan programnya dan contoh keluarannya. Sertakan pula komentar anda tentang cara kerja metode Bisection meliputi keakuratan hasil dan waktu prosesnya.

False Position Method (Metode Posisi Salah)

To check your root substitute your estimate back into the original equation – see if it is satisfied.

Kelemahan metode bisection adalah dalam membagi interval x_1 dan x_2 dalam paruh yang sama tidak ada perhitungan mengenai besar $f(x_1)$ dan $f(x_2)$. Dengan penjelasan grafik dapat diturunkan suatu formula yang diberi nama metode interpolasi linier atau dengan nama lain metode posisi salah

- Dengan memakai segitiga, perpotongan garis lurus dengan sumbu x dapat ditaksir sebagai :



$$\frac{f(x_1)}{x' - x_1} = \frac{f(x_2)}{x' - x_2}$$

Buktikan untuk PR

yang dapat diselesaikan

$$x' = x_2 - \frac{f(x_2)(x_1 - x_2)}{f(x_1) - f(x_2)}$$

persamaan ini digunakan untuk menggantikan persamaan pada langkah 2 algoritma metode bisection

TUGAS PRAKTIKUM

Cara dan tugas yang sama seperti pada Metode Bisection

```
ALGORITMA Metode_Akolade;
DEKLARASI
    x1,x2,xr,xn,akar: real;
    AA: real;
    ES,EA:real;
    i,IM,iterasi : integer;
    function FngsPers(x:real): real;
    function MtdAkolade(xl,xu:real): real;
DESKRIPSI
  read(x1); read(x2);            {nilai awal yg mengurung akar}
  read(ES);                                 {estimasi kesalahan}
  read(IM);                                 {iterasi maks}
  if FngsPers(x1)*FngsPers(x2)>0 then
       write('Tebakan X1 dan X2 tidak mengurung akar')
    else
        xr:= MtdAkolade(x1,x2);
        for i:=2 to IM do
            AA:=FngsPers(x1)*FngsPers(xr);
            if AA=0 then
                  akar:=xr; EA:=0;
                  iterasi:=i; i:=IM;
            if AA<0 then x2:=xr;
            if AA>0 then x1:=xr;
            xn:= MtdAkolade(x1,x2);
            if xn=0 then xr:=xn
              else
                  EA:=abs((xn-xr)/xn)*100;
                  if EA<ES then
                        iterasi:=i; i:=IM;
                  xr:=xn;
            akar:=xr;
        endfor
        write(akar:5:6);                    {cetak hasil akar}
        if iterasi<>IM then
           write(iterasi);                  {cetak jumlah iterasi}
        write(EA:5:6);                      {cetak error aproksimasi}
```