Fakultas Informatika
School of Computing
Telkom University

CNH2B4 / KOMPUTASI NUMERIK

TIM DOSEN

KK MODELING AND COMPUTATIONAL EXPERIMENT

2 SOLUSI PERSAMAAN NONLINEAR: METODE BAGIDUA & REGULA FALSI

SOLUSI PERSAMAAN NONLINEAR

- Metode Bagidua
- Metode Regula Falsi (False Position)
- Metode Iterasi Titik-Tetap
- Metode Newton-Raphson
- Metode Secant

- Metode Tertutup
  - Diberikan selang yang sudah diketahui memiliki akar.
  - Iterasi yang dilakukan dalam selang ini dipastikan konvergen menuju suatu akar.
  - Metode Bagidua dan Metode Regula Falsi termasuk metode tertutup.
- Metode Terbuka
  - Diperlukan tebakkan awal akar untuk memulai iterasi pencarian akar.
  - Tebakkan awal akar yang tidak baik dapat menyebabkan iterasinya divergen.
  - Metode Iterasi Titik Tetap, Metode Newton-Raphson, dan metode Secant termasuk metode tertutup.

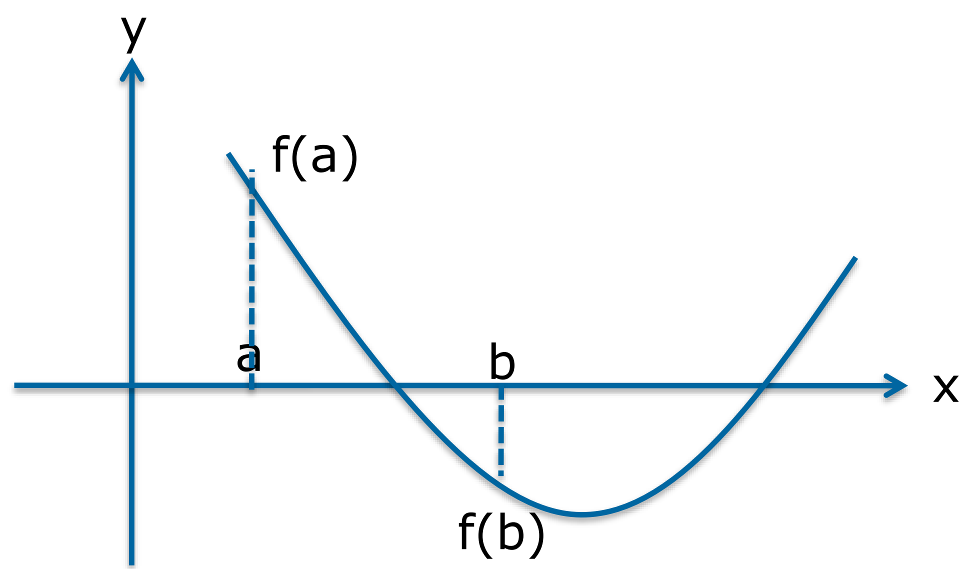
PERSAMAAN NONLINEAR

- Contoh persamaan-persamaan nonlinear:

1. $f(x) = x^2 + 2x - 4$
2. $f(x) = x^3 + 3x^2 - 4x + 5$
3. $f(x) = x^2 + \sin x$
4. $f(x) = x^4 - 2x \cos x$
5. $f(x) = 2x^3 + e^{2x}$
6. $f(x) = \log(x) + x^3 - 2\cos x$
7. $f(x) = e^{2x} + x^2 - 2\sinh x$

- Solusi dari persamaan nonlinear adalah mencari nilai x sedemikian sehingga *f*(*x*) = 0 atau biasa disebut dengan mencari akar dari persamaan nonlinear f(x).

Syarat Cukup Keberadaan Akar



$f(a) * f(b) < 0$

Jika *f(a)*f(b)< 0* dan *f(x)* kontinu dalam selang [*a*,*b*], maka paling sedikit terdapat satu buah akar persamaan *f(x) = 0* di dalam selang [*a*,*b*].

METODE BAGIDUA

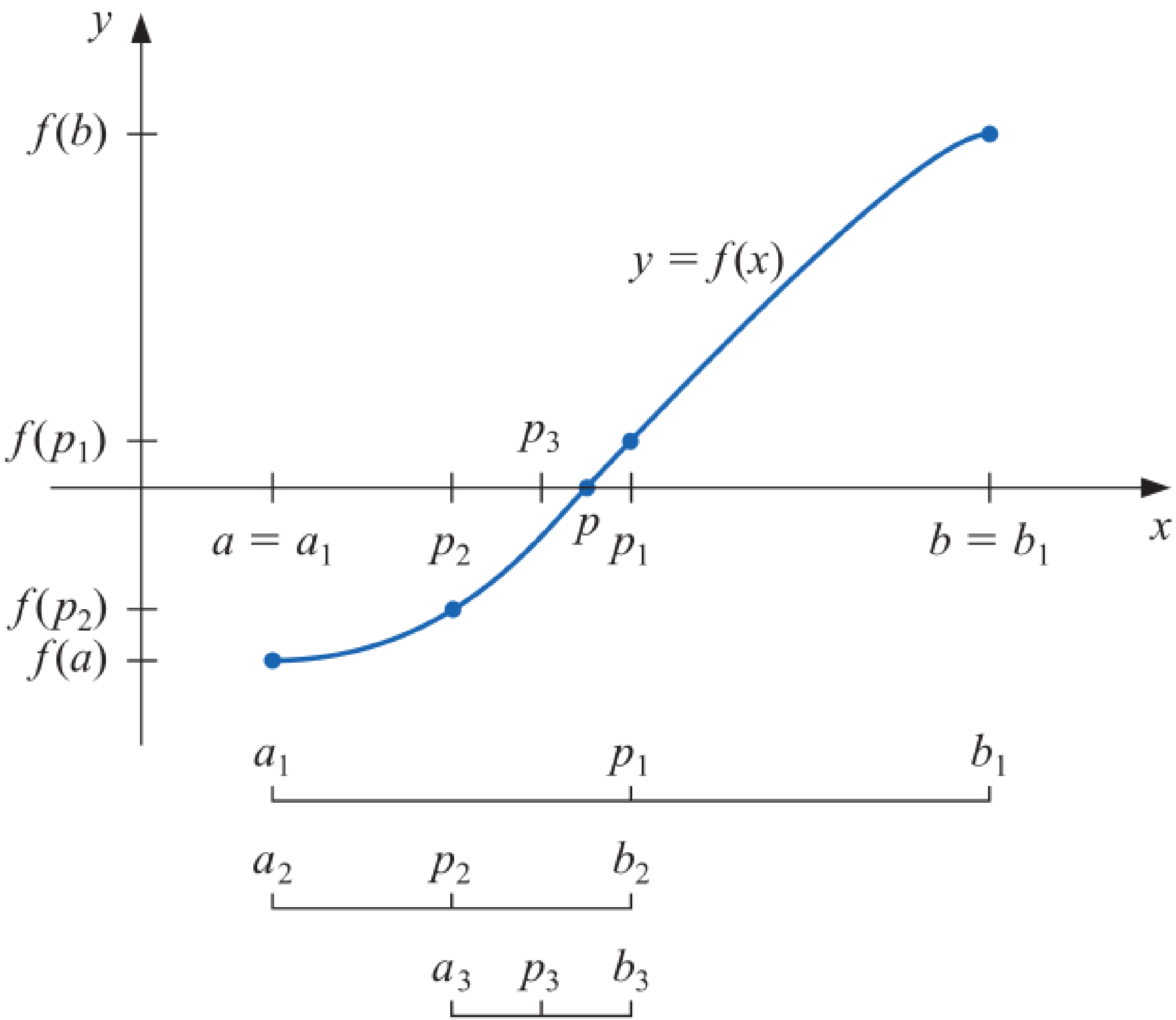
Misalkan f(x) adalah fungsi kontinu dalam selang [a,b] dengan f(a) dan f(b) berlawanan tanda. Misalkan p adalah akar dari f(x).

- Lakukan $a_1 = a$, dan $b_1 = b$, dan misalkan p_1 adalah titik tengah dari [a,b], yaitu $p_1 = a_1 + (b_1 - a_1)/2 = (a_1 + b_1)/2$.
- Jika $f(p_1) = 0$, maka $p = p_1$, dan kita selesai.
- Jika $f(p_1) \neq 0$ maka $f(p_1)$ mempunyai tanda yang sama dengan $f(a_1)$ atau $f(b_1)$:

- jika $f(p_1)$ dan $f(a_1)$ mempunyai tanda yang sama, maka $p \in (p_1,b_1)$. Lakukan $a_2 = p_1$, dan $b_2 = b_1$.
- jika $f(p_1)$ dan $f(a_1)$ berlawanan tanda, maka $p \in (a_1,p_1)$. Lakukan $a_2 = a_1$, dan $b_2 = p_1$.

- Lakukan proses yang sama untuk selang $[a_2,b_2]$.

METODE BAGIDUA



Burden, Richard L., and J. Douglas Faires. *Numerical Analysis*. Brooks/Cole, *USA*, 2001.

Algoritma Metode Bagidua

Bisection

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs:

INPUT endpoints a, b; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$;
$FA = f(a)$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = a + (b - a)/2$; (*Compute* p_i.)
$FP = f(p)$.

Step 4 If $FP = 0$ or $(b - a)/2 < TOL$ then
OUTPUT (p); (*Procedure completed successfully.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 If $FA \cdot FP > 0$ then set $a = p$; (*Compute* a_i, b_i.)
$FA = FP$
else set $b = p$. (*FA is unchanged.*)

Step 7 OUTPUT ('Method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP.

Burden, Richard L., and J. Douglas Faires. *Numerical Analysis*. Brooks/Cole, *USA*, 2001.

Contoh Kasus (1)

Tentukan akar dari f(x) = x^3 - 2x^2 – 5 dalam selang [2,3] menggunakan metode Bagidua.

n	a_n	b_n	p_n	f(a_n)	f(b_n)	f(p_n)	error
1	2.000000	3.000000	2.500000	-5.000000	4.000000	-1.875000	0.500000
2	2.500000	3.000000	2.750000	-1.875000	4.000000	0.671875	0.250000
3	2.500000	2.750000	2.625000	-1.875000	0.671875	-0.693359	0.125000
4	2.625000	2.750000	2.687500	-0.693359	0.671875	-0.034424	0.062500
...	...	...	...	...	...	...	...
15	2.690613	2.690674	2.690643	-0.000380	0.000289	-0.000045	0.000031
16	2.690643	2.690674	2.690659	-0.000045	0.000289	0.000122	0.000015
17	2.690643	2.690659	2.690651	-0.000045	0.000122	0.000038	0.000008

akar dari f(x) = x^3 - 2x^2 - 5 adalah x = 2.690651

Contoh Kasus (2)

Tentukan akar dari f(x) = sin(x) – exp(-x) dalam selang [3,4] menggunakan metode Bagidua.

n	a_n	b_n	p_n	f(a_n)	f(b_n)	f(p_n)	error
1	3.000000	4.000000	3.500000	0.091333	-0.775118	-0.380981	0.500000
2	3.000000	3.500000	3.250000	0.091333	-0.380981	-0.146969	0.250000
3	3.000000	3.250000	3.125000	0.091333	-0.146969	-0.027345	0.125000
4	3.000000	3.125000	3.062500	0.091333	-0.027345	0.032240	0.062500
...							
16	3.096344	3.096375	3.096359	0.000019	-0.000010	0.000004	0.000015
17	3.096359	3.096375	3.096367	0.000004	-0.000010	-0.000003	0.000008

akar dari f(x) = sin(x) – exp(-x) adalah x = 3.096367

Metode Regula Falsi/ *False Position*

- Misalkan $f(x)$ fungsi yang kontinu pada selang [p0,p1] dengan f(p0) berbeda tanda dengan f(p1). Misalkan p adalah akar dari f(x).
- Pertama, buat garis lurus yang menghubungkan titik (*p0*,*f*(*p0*)) dengan titik (*p1*, *f*(*p1*)).

$$y - f(p_1) = \frac{f(p_1) - f(p_0)}{p_1 - p_0}\left(x - p_1\right)$$

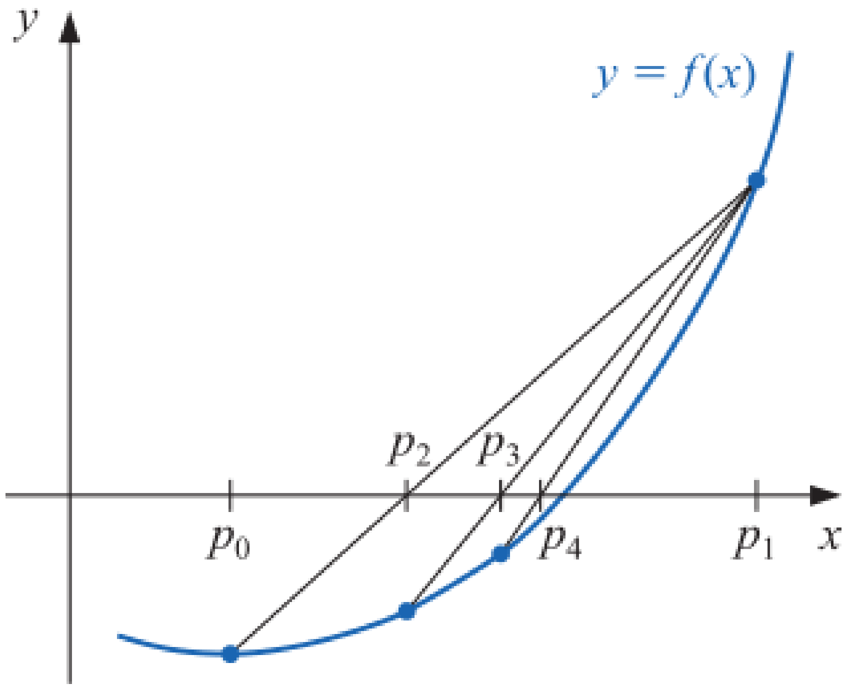
- Perpotongan garis tersebut dengan sumbu-x menghasilkan hampiran akar. Subsitusi x = p dan y = 0, maka

$$p = p_1 - \frac{f(p_1)\left(p_1 - p_0\right)}{f(p_1) - f(p_0)}$$

Metode Regula Falsi/ *False Position*

$$p_2 = p_1 - \frac{f(p_1)(p_1 - p_0)}{f(p_1) - f(p_0)}$$

$$p_3 = p_2 - \frac{f(p_2)(p_2 - p_1)}{f(p_2) - f(p_1)}$$



Burden, Richard L., and J. Douglas Faires. *Numerical Analysis*. Brooks/Cole, *USA*, 2001.

Algoritma False Position

False Position

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[p_0, p_1]$ where $f(p_0)$ and $f(p_1)$ have opposite signs:

INPUT initial approximations p_0, p_1; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 2$;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$.

Step 2 While $i \leq N_0$ do Steps 3–7.

Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$. (*Compute p_i.*)

Step 4 If $|p - p_1| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$;
$q = f(p)$.

Step 6 If $q \cdot q_1 < 0$ then set $p_0 = p_1$;
$q_0 = q_1$.

Step 7 Set $p_1 = p$;
$q_1 = q$.

Step 8 OUTPUT ('Method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure unsuccessful.*)
STOP. ■

Burden, Richard L., and J. Douglas Faires. *Numerical Analysis*. Brooks/Cole, *USA*, 2001.

Contoh Kasus (1)

Tentukan akar dari f(x) = x^3 - 2x^2 – 5 dalam selang [2,3] menggunakan metode Regula Falsi.

n	p_n	f(p_n)	error
0	2.000000	-5.000000	
1	3.000000	4.000000	
2	2.555556	-1.371742	0.444444
3	2.669050	-0.233802	0.113494
4	2.687326	-0.036323	0.018276
5	2.690140	-0.005560	0.002814
6	2.690570	-0.000849	0.000430
7	2.690636	-0.000130	0.000066
8	2.690646	-0.000020	0.000010
9	2.690647	-0.000003	0.000002

akar dari f(x) = x^3 - 2x^2 - 5 adalah x = 2.690647

Contoh Kasus (2)

Tentukan akar dari f(x) = sin(x) – exp(-x) dalam selang [3,4] menggunakan metode Regula Falsi.

n	p_n	f(p_n)	error
0	3.000000	0.091333	
1	4.000000	-0.775118	
2	3.105410	-0.008632	0.894590
3	3.096308	0.000053	0.009102
4	3.096364	0.000000	0.000056
5	3.096364	0.000000	0.000000

akar dari f(x) = sin(x) - exp(-x) adalah x = 3.096364

Fakultas Informatika
School of Computing
Telkom University

THANK YOU