



TSK21275 METODE NUMERIK PROGRAM STUDI TEKNIK KOMPUTER UNIVERSITAS DIPONEGORO

6. Sistem Persamaan Nonlinear

Eko Didik Widiyanto (didik@live.undip.ac.id)

Pendahuluan

- Dalam dunia nyata, umumnya persamaan matematika dapat lebih dari satu, sehingga membentuk sebuah sistem yang disebut sistem persamaan nirlanjar atau nonlinear
- **Bentuk umum:** $f_1(x_1, x_2, \dots, x_n) = 0$
 $f_2(x_1, x_2, \dots, x_n) = 0$
...
 $f_n(x_1, x_2, \dots, x_n) = 0$
- Solusi sistem ini adalah himpunan nilai x simultan, x_1, x_2, \dots, x_n , yang memenuhi seluruh persamaan
- **Metode:** iterasi titik tetap atau Newton-Raphson
- **Referensi**
[1] Rinaldi Munir, Metode Numerik. Penerbit Informatika, 2015 (Bab 3.7). Paparan ini sebagian besar mengambil dari slide paparan buku

1. Metode iterasi titik-tetap

- Prosedur iterasi titik-tetap untuk sistem dengan dua persamaan nirlanjar:

$$x_{r+1} = g_1(x_r, y_r)$$

$$y_{r+1} = g_2(x_r, y_r) \text{ untuk } r = 0, 1, 2, \dots$$

- Metode iterasi titik-tetap ini dinamakan metode **iterasi Jacobi**
- Kondisi berhenti (konvergen) adalah $|x_{r+1} - x_r| < \varepsilon$ dan $|y_{r+1} - y_r| < \varepsilon$
- Kecepatan konvergensi masih dapat ditingkatkan

Perbaikan kecepatan konvergensi

- Kecepatan konvergensi lelaran titik-tetap ini dapat ditingkatkan. Nilai x_{r+1} yang baru dihitung langsung digunakan untuk menghitung y_{r+1}

$$x_{r+1} = g_1(x_r, y_r)$$


$$y_{r+1} = g_2(x_{r+1}, y_r) \text{ untuk } r = 0, 1, 2, \dots$$

- Metode iterasi titik-tetap ini → metode **iterasi Seidel**

- Kondisi berhenti adalah $|x_{r+1} - x_r| < \varepsilon$ dan $|y_{r+1} - y_r| < \varepsilon$

- Iterasi Seidel untuk fungsi tiga persamaan nonlinear:

$$x_{r+1} = g_1(x_r, y_r, z_r) \quad , \quad y_{r+1} = g_2(x_{r+1}, y_r, z_r), \quad z_{r+1} = g_3(x_{r+1}, y_{r+1}, z_r), \quad r=0, 1, 2, \dots$$

- Kondisi berhenti: $|x_{r+1} - x_r| < \varepsilon$ dan $|y_{r+1} - y_r| < \varepsilon$ dan $|z_{r+1} - z_r| < \varepsilon$

Contoh

- Selesaikan sistem persamaan non-linear berikut:

$$f_1(x, y) = x^2 + xy - 10 = 0$$

$$f_2(x, y) = y + 3xy^2 - 57 = 0$$

Tebakan $x_0 = 1.5$, $y_0 = 3.5$, dan epsilon 0.000001

- Penyelesaian:** Terdapat beberapa kemungkinan prosedur iterasi yang dapat dibentuk (iterasi Seidel)



$$x_{r+1} = \frac{10 - x_r^2}{y_r}$$

$$y_{r+1} = 57 - 3x_{r+1}y_r^2$$

i	x	y	deltaX	deltaY
0	1.500000	3.500000	0.000000	0.000000
1	2.214286	-24.374992	0.714286	27.874992
2	-0.209105	429.713562	2.423391	454.088562
3	0.023170	-12778.040039	0.232275	13207.753906
4	-0.000783	383377.625000	0.023952	396155.656250

....

- Solusi divergen !!

Solusi (2)

■ Penyelesaian:

$$x_{r+1} = \sqrt{10 - x_r y_r}$$

$$y_{r+1} = \sqrt{\frac{57 - y_r}{3 x_{r+1}}}$$



Pemilihan persamaan
iterasi membuat
solusi konvergen

r	x	y	deltaX	deltaY
0	1.500000	3.500000	0.000000	0.000000
1	2.179449	2.860506	0.679449	0.639494
2	1.940534	3.049551	0.238916	0.189045
3	2.020456	2.983405	0.079922	0.066146
4	1.993028	3.005704	0.027428	0.022300
5	2.002385	2.998054	0.009357	0.007650
6	1.999185	3.000666	0.003200	0.002611
7	2.000279	2.999773	0.001094	0.000893
8	1.999905	3.000078	0.000374	0.000305
9	2.000033	2.999973	0.000128	0.000104
10	1.999989	3.000009	0.000044	0.000036
11	2.000004	2.999997	0.000015	0.000012
12	1.999999	3.000001	0.000005	0.000004
13	2.000000	3.000000	0.000002	0.000001

■ Ditemukan akar sejadi $x=2$ dan $y=3$

Kode C

```
1  /*
2  * iterasi Seidel non-linear
3  */
4
5  #include <stdio.h>
6  #include <stdlib.h>
7  #include <math.h>
8
9  #define MAX_ITERATION 1000
10
11 int main() {
12     const float x0 = 1.5;
13     const float y0 = 3.5;
14     const float epsilon = 0.000001;
15
16     int i = 0;
17     double x[MAX_ITERATION];
18     double y[MAX_ITERATION];
19     float deltaX = 0, deltaY = 0;
20
21     x[0] = x0; y[0] = y0;
22
23     printf("-----\n");
24     printf("r\t x\t y\t deltaX\t deltaY\n");
25     printf("-----\n");
26     do {
27         if (i >= MAX_ITERATION) break;
28         //x[i+1] = (10-x[i] * x[i])/y[i];
29         //y[i+1] = 57 - (3*x[i+1] * y[i] * y[i]);
30         x[i+1] = sqrt(10 - x[i]*y[i]);
31         y[i+1] = sqrt((57-y[i])/(3*x[i+1]));
32
33         printf("%d\t %f\t %f\t %f\t %f\n", i, x[i], y[i], deltaX, deltaY);
34         deltaX = fabs(x[i+1] - x[i]);
35         deltaY = fabs(y[i+1] - y[i]);
36         i++;
37     } while (deltaX > epsilon || deltaY > epsilon );
38 }
39
```

Syarat perlu konvergen

- Konvergensi metode iterasi titik tetap tergantung pada bentuk persamaan prosedur iterasi dan tebakan awal
- Syarat perlu konvergen untuk sistem dua persamaan

$$\left| \frac{\partial g_1}{\partial x} \right| + \left| \frac{\partial g_1}{\partial y} \right| < 1 \quad \text{dan} \quad \left| \frac{\partial g_2}{\partial x} \right| + \left| \frac{\partial g_2}{\partial y} \right| < 1$$

- di dalam selang yang mengandung titik tetap (p,q)
- **Latihan:** analisis kekonvergenan dari dua persamaan prosedur iterasi di contoh sebelumnya

2. Metode Newton Raphson

- Penurunan metode Newton Raphson dengan deret Taylor:

$$f(x_{r+1}) = f(x_r) + (x_{r+1} - x_r) f'(x_r)$$

- Untuk fungsi dengan dua peubah: $f_1(x,y) = u$ dan $f_2(x,y) = v$

$$u_{r+1} = u_r + (x_{r+1} - x_r) \frac{\partial u_r}{\partial x} + (y_{r+1} - y_r) \frac{\partial u_r}{\partial y}$$

$$v_{r+1} = v_r + (x_{r+1} - x_r) \frac{\partial v_r}{\partial x} + (y_{r+1} - y_r) \frac{\partial v_r}{\partial y}$$

- Untuk persoalan pencarian akar, maka $u_{r+1} = v_{r+1} = 0$

$$x_{r+1} \frac{\partial u_r}{\partial x} + y_{r+1} \frac{\partial u_r}{\partial y} = -u_r + x_r \frac{\partial u_r}{\partial x} + y_r \frac{\partial u_r}{\partial y}$$

$$x_{r+1} \frac{\partial v_r}{\partial x} + y_{r+1} \frac{\partial v_r}{\partial y} = -v_r + x_r \frac{\partial v_r}{\partial x} + y_r \frac{\partial v_r}{\partial y}$$

Determinan Jacobi

$$x_{r+1} \frac{\partial u_r}{\partial x} + y_{r+1} \frac{\partial u_r}{\partial y} = -u_r + x_r \frac{\partial u_r}{\partial x} + y_r \frac{\partial u_r}{\partial y}$$

$$x_{r+1} \frac{\partial v_r}{\partial x} + y_{r+1} \frac{\partial v_r}{\partial y} = -v_r + x_r \frac{\partial v_r}{\partial x} + y_r \frac{\partial v_r}{\partial y}$$

- Dengan penyelesaian aljabar:

$$x_{r+1} = x_r - \frac{u_r \frac{\partial v_r}{\partial y} + v_r \frac{\partial v_r}{\partial y}}{\frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x}}$$

$$y_{r+1} = y_r + \frac{u_r \frac{\partial v_r}{\partial x} - v_r \frac{\partial u_r}{\partial x}}{\frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x}}$$

- Penyebut dari tiap persamaan diacu sebagai **determinan Jacobi** dari sistem tersebut
- Metode Newton-Raphson dapat digeneralisasi untuk sistem dengan n persamaan

Contoh

- Selesaikan sistem persamaan non-linear berikut:

$$f_1(x, y) = u = x^2 + xy - 10 = 0$$

$$\text{Tebakan } x_0 = 1.5, y_0 = 3.5$$

$$f_2(x, y) = v = y + 3xy^2 - 57 = 0$$

- **Penyelesaian:** $\frac{\partial u_0}{\partial x} = 2x + y = 2(1,5) + 3,5 = 6,5$ $\frac{\partial u_0}{\partial y} = x = 1,5$

$$\frac{\partial v_0}{\partial x} = 3y^2 = 3(3,5)^2 = 36,75 \quad \frac{\partial v_0}{\partial y} = 1 + 6xy = 1 + 6(1,5)(3,5) = 32,5$$

- Determinan Jacobi untuk iterasi pertama

$$\det = \frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x} = 6,5(32,5) - 1,5(36,75) = 156,125$$

- Nilai fungsi u_0 dan v_0 dari tebakan awal:

$$u_0 = (1,5)^2 + 1,5(3,5) - 10 = -2,5$$

$$v_0 = (3,5) + 3(1,5)(3,5)^2 - 57 = 1,625$$

Contoh - cont'd

$$x_{r+1} = x_r - \frac{u_r \frac{\partial v_r}{\partial y} + v_r \frac{\partial v_r}{\partial y}}{\frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x}} \quad y_{r+1} = y_r + \frac{u_r \frac{\partial v_r}{\partial x} - v_r \frac{\partial u_r}{\partial x}}{\frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x}}$$

- Nilai x dan y pada iterasi pertama:

$$x_1 = 1,5 - \frac{(-2,5)(32,5) - 1,625(1,5)}{156,125} = 2,03603$$

$$y_1 = 3,5 + \frac{(-2,5)(36,75) - 1,625(6,5)}{156,125} = 2,84388$$

- Jika diteruskan, iterasi konvergen ke akar sejati $x=2$ dan $y=3$

Contoh - hasil

r	x	y	deltaX	deltaY
0	1.500000	3.500000	0.000000	0.000000
1	1.398458	5.606684	0.101542	2.106684
2	1.848466	3.048090	0.450009	2.558594
3	1.996500	3.021253	0.148033	0.026837
4	1.996880	3.010930	0.000380	0.010323
5	1.998392	3.005629	0.001512	0.005301
6	1.999171	3.002903	0.000779	0.002726
7	1.999572	3.001498	0.000401	0.001405
8	1.999779	3.000773	0.000207	0.000725
9	1.999886	3.000399	0.000107	0.000374
10	1.999941	3.000206	0.000055	0.000193
11	1.999970	3.000107	0.000028	0.000100
12	1.999984	3.000055	0.000015	0.000052
13	1.999992	3.000028	0.000008	0.000027
14	1.999996	3.000015	0.000004	0.000014
15	1.999998	3.000008	0.000002	0.000007
16	1.999999	3.000004	0.000001	0.000004
17	1.999999	3.000002	0.000001	0.000002

- Akar ditemukan setelah 17 iterasi

Contoh - kode

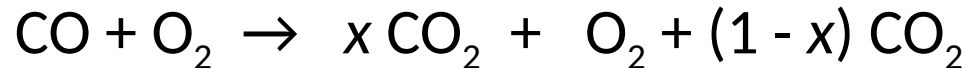
```
24 printf("-----\n");
25 printf("r\t x\t y\t deltaX\t deltaY\n");
26 printf("-----\n");
27 do {
28     if (i >= MAX_ITERATION) break;
29     // u = x^2+xy-10 --> dux = 2x+y, duy = x
30     // v = y+3xy^2-57 --> dvx = 3y^2, dvy = 1+6xy
31     u[i] = x[i]*x[i] + x[i]*y[i] - 10;
32     v[i] = y[i]+3*x[i]*y[i]*y[i] - 57;
33     dux = 2*x[i] + y[i]; duy = x[i];
34     dvx = 3*y[i]*y[i]; dvy = 1+6*x[i]*y[i];
35     // Jacobi determinan = dux.dvy - duy.dvx
36     det = dux * dvy - duy * dvx;
37     // Iteration equation
38     x[i+1] = x[i] - (u[i]*dvy-v[i]*duy) / det;
39     y[i+1] = y[i] + (u[i]*dvx-v[i]*dux) / det;
40
41     printf("%d\t %f\t %f\t %f\t %f\n", i, x[i], y[i], deltaX, deltaY);
42     deltaX = fabs(x[i+1] - x[i]);
43     deltaY = fabs(y[i+1] - y[i]);
44     i++;
45 } while (deltaX > epsilon || deltaY > epsilon );
46
47
```

Contoh penerapan

Dalam suatu proses Teknik Kimia, campuran karbon monoksida dan oksigen mencapai kesetimbangan pada suhu 300°K dan tekanan 5 atm. Reaksi teoritisnya adalah



Reaksi kimia yang sebenarnya terjadi dapat ditulis sebagai



Persamaan kesetimbangan kimia untuk menentukan fraksi mol CO yang tersisa, yaitu x , ditulis sebagai

$$K_p = \frac{(1-x)(3+x)^{1/2}}{x(x+1)^{1/2} p^{1/2}}; 0 < x < 1$$

yang dalam hal ini, $K_p = 3,06$ adalah tetapan kesetimbangan untuk reaksi $\text{CO} + 1/2 \text{O}_2$ pada 3000°K dan $P = 5 \text{ atm}$. Tentukan nilai x dengan metode regula falsi yang diperbaiki.

Penyelesaian

Penyelesaian: Persoalan ini memang lebih tepat diselesaikan dengan metode tertutup karena x adalah fraksi mol yang nilainya terletak antara 0 dan 1.

Fungsi yang akan dicari akarnya dapat ditulis sebagai

$$f(x) = \frac{(1-x)(3+x)^{1/2}}{x(x+1)^{1/2} p^{1/2}} - K_p; 0 < x < 1$$

dengan $K_p = 3.06$ dan $P = 5 \text{ atm}$.

Selang yang mengandung akar adalah $[0.1, 0.9]$. Nilai fungsi di ujung-ujung selang adalah $f(0.1) = 3.696815$ dan $f(0.9) = -2.988809$ yang memenuhi $f(0.1) f(0.9) < 0 \rightarrow$ Syarat cukup mempunyai akar

Tabel iterasi

r	a	c	b	$f(a)$	$f(c)$	$f(b)$	Selang baru	Lebar nya
0	0.100000	0.542360	0.900000	3.696815	-2.488120	-2.988809	[a, c]	0.442360
1	0.100000	0.288552	0.542360	1.848407	-1.298490	-2.488120	[a, c]	0.188552
2	0.100000	0.178401	0.288552	0.924204	0.322490	-1.298490	[c, b]	0.110151
3	0.178401	0.200315	0.288552	0.322490	-0.144794	-1.298490	[a, c]	0.021914
4	0.178401	0.193525	0.200315	0.322490	-0.011477	-0.144794	[a, c]	0.015124
5	0.178401	0.192520	0.193525	0.161242	0.009064	-0.011477	[c, b]	0.001005
6	0.192520	0.192963	0.193525	0.009064	-0.000027	-0.011477	[a, c]	0.000443
7	0.192520	0.192962	0.192963	0.009064	-0.000000	-0.000027	[a, c]	0.000442

- Hampiran akarnya adalah $x=0,192962$
- Jadi, setelah reaksi berlangsung, fraksi mol CO yang tersisa adalah 0,192962