Tugas Modul I

Solusi Persamaan Nonlinear

Dibuat Oleh :

Nashirudin Baqiy

24060119130045

Asisten Praktikum :

Muhammad Rizqi Arya Pradana

Ibnu Nahwitama

DEPARTEMEN INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNVERSITAS DIPONEGORO

2020

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Tujuan

1. Dapat menghitung akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson dan metode Secant
2. Mencari besarnya kesalahan dari suatu perhitungan akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson, dan metode Secant

1.2 Rumusan Permasalahan

1. Diberikan fungsi $f(x) = e\ x + x\ 2 - 3x - 2 = 0$ terdapat sebuah akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Biseksi dengan toleransi kesalahan 1e-5
2. Diberikan fungsi $f(x) = e\ x + x\ 2 - 3x - 2 = 0$ mempunyai akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Newton Raphson dengan toleransi kesalahan 1e-5.

BAB II
DASAR TEORI

2.1 Pendahuluan Teori

Pencarian akar(penyelesaian) suatu persamaan non-linear, $y = f(x)$, adalah mencari suatu harga $x *$, yang apabila disubstitusikan ke dalam persamaan itu, akan memberikan harga fungsi nol. Secara matematika

$$f(x *) = 0$$

Sebagai contoh, akar dari persamaan $f(x) = x - e^{\frac{1}{x}}$ adalah 1.763223 karena, apabila harga $x *$ = 1.763223 disubstitusikan ke dalam persamaan itu, harga fungsi itu menjadi nol. Metode-metode numeric yang banyak digunakan untuk menyelesaikan persamaan non-linear adalah metode biseksi, metode regula falsi, secant, newton -Raphson dan titik tetap. Tiga metode akan dibicarakan di sini adalah metode biseksi , metode Newton-Raphson dan metode Secant

2.2 Metode Biseksi

Dalam metode Biseksi, interval yang mengandung akar dibagi menjadi dua secara berurutan hingga ukuran interval mengecil dan akhirnya mencapai harga toleransi kesalahan yang diinginkan. Dalam interval [a,b] terdapat sebuah akar (yang akan dicari), apabila dipenuhi :

$$f(a) * f(b) \leq 0$$

Proses :

1. iter=0
2. $x_l = a, x_r = b$
3. Jika $fx(x_l) * fx(x_m) < 0$, kerjakan langkah 4-langkah 7
4. $x_m = 0.5 * (x_l + x_r)$
5. $iter = iter + 1$
6. $delta = |x_m - x_l|$
7. Selagi $delta > tol\&iter < maxit$kerjakan :

   7.1. jika $fx(x_l) * fx(x_m) < 0$, maka

   $$x_r = x_m$$

   Jika tidak, $x_l = x_m$

   7.2. baharui harga x_m: $x_m = 0.5 * (x_l + x_r)$
8. Akar pendekatan = x_m
9. Selesai

2.3 Metode Newton-Raphson

Metode Newton Raphson, dalam mencari akar suatu fungsi nonlinear $y = f(x)$ memerlukan evaluasi harga fungsi dan turunannya pada sembarang titik x yang merupakan harga awal tebakan akar fungsi tersebut. Metode ini didasarkan atas perluasan deret Taylor di sekitar suatu titik. Karenanya, apabila harga awal tebakan jauh dari akar sebenarnya, konvergensi akan lambat atau mungkin tidak dicapai sama sekali.

Proses :

1. iter=0
2. Hitung $fx(x_c)$ dan $dfx(x_c)$
3. iter=iter+1
4. $x_{c+1} = x_c - \frac{fx(x_c)}{dfx(x_c)}$
5. $delta = |x_{c+1} - x_c|$
6. Jika$delta \leq eps$& iter>maxit maka Akar pendekatan = x_{c+1}, Selesai
7. $x_c = x_{c+1}$, kembali ke langkah 2

BAB III
PEMBAHASAN

3.1 Penyelesaian dengan Metode Biseksi

Syarat dari Metode Biseksi ialah $f(x_l) * f(x_m) \leq 0$ maka $f(-1) * f(1)$ harus $\leq\ 0$. Selanjut-selanjutnya $f(x_l) * f(x_m)$ disebut dengan check ditabel.

3.1.1 Dengan Excel

Isi

Iterasi	xl	xr	xm	delta	f(xl)	f(xm)	check
1	-1	1	=(B6+C6)/2	=ABS(B6-C6)	=(EXP(B6)+(B6)^2-3*	=(EXP(D6)+(D6)^2-3*	=F6*G6
2	=IF(H6<0;B6;D6)	=IF(H6<0;D6;C6)	=(B7+C7)/2	=ABS(B7-C7)	=(EXP(B7)+(B7)^2-3*	=(EXP(D7)+(D7)^2-3*	=F7*G7
3	=IF(H7<0;B7;D7)	=IF(H7<0;D7;C7)	=(B8+C8)/2	=ABS(B8-C8)	=(EXP(B8)+(B8)^2-3*	=(EXP(D8)+(D8)^2-3*	=F8*G8
4	=IF(H8<0;B8;D8)	=IF(H8<0;D8;C8)	=(B9+C9)/2	=ABS(B9-C9)	=(EXP(B9)+(B9)^2-3*	=(EXP(D9)+(D9)^2-3*	=F9*G9
5	=IF(H9<0;B9;D9)	=IF(H9<0;D9;C9)	=(B10+C10)/2	=ABS(B10-C10)	=(EXP(B10)+(B10)^2-	=(EXP(D10)+(D10)^2-	=F10*G10
6	=IF(H10<0;B10;D10)	=IF(H10<0;D10;C10)	=(B11+C11)/2	=ABS(B11-C11)	=(EXP(B11)+(B11)^2-	=(EXP(D11)+(D11)^2-	=F11*G11
7	=IF(H11<0;B11;D11)	=IF(H11<0;D11;C11)	=(B12+C12)/2	=ABS(B12-C12)	=(EXP(B12)+(B12)^2-	=(EXP(D12)+(D12)^2-	=F12*G12
8	=IF(H12<0;B12;D12)	=IF(H12<0;D12;C12)	=(B13+C13)/2	=ABS(B13-C13)	=(EXP(B13)+(B13)^2-	=(EXP(D13)+(D13)^2-	=F13*G13
9	=IF(H13<0;B13;D13)	=IF(H13<0;D13;C13)	=(B14+C14)/2	=ABS(B14-C14)	=(EXP(B14)+(B14)^2-	=(EXP(D14)+(D14)^2-	=F14*G14
10	=IF(H14<0;B14;D14)	=IF(H14<0;D14;C14)	=(B15+C15)/2	=ABS(B15-C15)	=(EXP(B15)+(B15)^2-	=(EXP(D15)+(D15)^2-	=F15*G15
11	=IF(H15<0;B15;D15)	=IF(H15<0;D15;C15)	=(B16+C16)/2	=ABS(B16-C16)	=(EXP(B16)+(B16)^2-	=(EXP(D16)+(D16)^2-	=F16*G16
12	=IF(H16<0;B16;D16)	=IF(H16<0;D16;C16)	=(B17+C17)/2	=ABS(B17-C17)	=(EXP(B17)+(B17)^2-	=(EXP(D17)+(D17)^2-	=F17*G17
13	=IF(H17<0;B17;D17)	=IF(H17<0;D17;C17)	=(B18+C18)/2	=ABS(B18-C18)	=(EXP(B18)+(B18)^2-	=(EXP(D18)+(D18)^2-	=F18*G18
14	=IF(H18<0;B18;D18)	=IF(H18<0;D18;C18)	=(B19+C19)/2	=ABS(B19-C19)	=(EXP(B19)+(B19)^2-	=(EXP(D19)+(D19)^2-	=F19*G19
15	=IF(H19<0;B19;D19)	=IF(H19<0;D19;C19)	=(B20+C20)/2	=ABS(B20-C20)	=(EXP(B20)+(B20)^2-	=(EXP(D20)+(D20)^2-	=F20*G20
16	=IF(H20<0;B20;D20)	=IF(H20<0;D20;C20)	=(B21+C21)/2	=ABS(B21-C21)	=(EXP(B21)+(B21)^2-	=(EXP(D21)+(D21)^2-	=F21*G21
17	=IF(H21<0;B21;D21)	=IF(H21<0;D21;C21)	=(B22+C22)/2	=ABS(B22-C22)	=(EXP(B22)+(B22)^2-	=(EXP(D22)+(D22)^2-	=F22*G22
18	=IF(H22<0;B22;D22)	=IF(H22<0;D22;C22)	=(B23+C23)/2	=ABS(B23-C23)	=(EXP(B23)+(B23)^2-	=(EXP(D23)+(D23)^2-	=F23*G23
19	=IF(H23<0;B23;D23)	=IF(H23<0;D23;C23)	=(B24+C24)/2	=ABS(B24-C24)	=(EXP(B24)+(B24)^2-	=(EXP(D24)+(D24)^2-	=F24*G24

Hasil

BISEKSI

$f(x)$ = e^x + x^2 − 3x − 2 = 0 selang [-1.0, 1.0] toleransi kesalahan 1e-5

f(-1) * f(1) = -3,03495 memenuhi syarat biseksi < 0

Iterasi	xl	xr	xm	delta	f(xl)	f(xm)	check
1	-1	1	0	1	2,367879	-1	-2,36788
2	-1	0	-0,5	0,5	2,367879	0,356531	0,844222
3	-0,5	0	-0,25	0,25	0,356531	-0,4087	-0,14571
4	-0,5	-0,25	-0,375	0,125	0,356531	-0,04709	-0,01679
5	-0,5	-0,375	-0,4375	0,0625	0,356531	0,149555	0,053321
6	-0,4375	-0,375	-0,40625	0,03125	0,149555	0,049933	0,007468
7	-0,40625	-0,375	-0,39063	0,015625	0,049933	0,001097	5,48E-05
8	-0,39063	-0,375	-0,38281	0,007813	0,001097	-0,02308	-2,5E-05
9	-0,39063	-0,38281	-0,38672	0,003906	0,001097	-0,01101	-1,2E-05
10	-0,39063	-0,38672	-0,38867	0,001953	0,001097	-0,00496	-5,4E-06
11	-0,39063	-0,38867	-0,38965	0,000977	0,001097	-0,00193	-2,1E-06
12	-0,39063	-0,38965	-0,39014	0,000488	0,001097	-0,00042	-4,6E-07
13	-0,39063	-0,39014	-0,39038	0,000244	0,001097	0,000339	3,72E-07
14	-0,39038	-0,39014	-0,39026	0,000122	0,000339	-4E-05	-1,4E-08
15	-0,39038	-0,39026	-0,39032	6,1E-05	0,000339	0,000149	5,06E-08
16	-0,39032	-0,39026	-0,39029	3,05E-05	0,000149	5,47E-05	8,17E-09
17	-0,39029	-0,39026	-0,39027	1,53E-05	5,47E-05	7,33E-06	4,01E-10
18	-0,39027	-0,39026	-0,39027	7,63E-06	7,33E-06	-1,6E-05	-1,2E-10

3.1.2 Dengan Python

Source Code

```
# BISECTION METHOD
# 24060119130045 - Nashirudin Baqiy

def bisect(f,a,b,tol=0.00001,maxiter=50):
    iterator = 0

    xl=a
    xr=b
    xm=(xl + xr) / 2

    if f(a)*f(b)==0:
      if f(a)==0:
            return a
      elif f(b)==0:
            return b
    elif f(a)*f(b)>0:
      raise ValueError('Nilai f(a) dan f(b) harus mempunyai
tanda yang berbeda')

    while iterator < maxiter:
      if f(xl)*f(xm)<0:
            xr = xm

      elif f(xl)*f(xm)>0:
            xl = xm

      elif f(xl)*f(xm)==0:
            return xm
```

```
        xm =(xl+xr)/2

        iterator=iterator+1

        delta=abs(xr-xl)

        if delta<tol:
            return xm

    return xm

def persamaan(x):
        import math
        return math.e**x + x**2 - 3*x - 2

print('Akarnya dengan Biseksi ='+str(bisect(persamaan,-1,1)))
```

Hasil

```
========== RESTART: F:/TUGAS/Praktikum/Metnum/Tugas1/py/biseksi.py ==========
Akarnya dengan Biseksi =-0.39026641845703125
>>>
```

3.1.3 Penjelasan

Kedua cara diatas baik excel maupun python menggunakan langkah-langkah yang sama yang telah dilampirkan di BAB II DASAR TEORI. Iter 0 yaitu mengecek apakah memenuhi syarat menggunakan metode biseksi atau tidak dengan cara $f(x_l) * f(x_m) < 0$. Jika memenuhi, maka masuk ke langkah 4 dst dari iter 1 selagi delta > tol & iter < maxit. Jika $f(x_l) * f(x_m) < 0$ pada iter 1, maka $x_r = x_m$ dan jika bukan $x_l = x_m$ kemudian diperbaru $x_m = (x_l + x_r)/2$ begitu juga dengan deltanya. Setelah stop maka didapat akarnya yaitu x_m.

Akar pendekatannya : -0.390266

Dengan Toleransi : 7.63e-6

3.2 Penyelesaian dengan Metode Newton-Raphson

3.2.1 Dengan Excel

Isi

iter	xc	f(xc)	delta	df(xc)
1	1	=EXP(B5)+B5^2-3*B5		=EXP(B5)+2*(B5)-3
2	=B5-C5/E5	=EXP(B6)+B6^2-3*B6	=ABS(B6-B5)	=EXP(B6)+2*(B6)-3
3	=B6-C6/E6	=EXP(B7)+B7^2-3*B7	=ABS(B7-B6)	=EXP(B7)+2*(B7)-3
4	=B7-C7/E7	=EXP(B8)+B8^2-3*B8	=ABS(B8-B7)	=EXP(B8)+2*(B8)-3
5	=B8-C8/E8	=EXP(B9)+B9^2-3*B9	=ABS(B9-B8)	=EXP(B9)+2*(B9)-3
6	=B9-C9/E9	=EXP(B10)+B10^2-3*	=ABS(B10-B9)	=EXP(B10)+2*(B10)-3

Hasil

Newton-Raphson

$f(x)$ = e^x + x^2 − 3x − 2 = 0 selang [-1.0, 1.0]

f'(x) = e^x + 2x - 3 toleransi kesalahan 1e-5

iter	xc	f(xc)	delta	df(xc)
1	1	-1,28172		1,718282
2	1,74593	1,541711	0,74593	6,22309
3	1,49819	0,223586	0,24774	4,469962
4	1,44817	0,008006	0,05002	4,15166
5	1,446241	1,16E-05	0,001928	4,139605
6	1,446239	2,46E-11	2,81E-06	4,139587

3.2.2 Dengan Python

Source Code

```
# NEWTON-RAPHSON METHOD
# 24060119130045 - Nashirudin Baqiy

from math import *

def newton_raphson(f,fdef,x0,eps=0.00001,maxiter=100):

    iterator=0

    while iterator < maxiter:
        xc = f(x0)
```

```
        xcdef = fdef(x0)

        iterator=iterator+1

        x1 = x0- (xc/xcdef)

        delta = abs(x1-x0)

        if delta<=eps:
            return x1

        x0 = x1

    return x0

def persamaan(x):
    return e**x + x**2 - 3*x - 2

def turunan(x):
    return e**x + 2*x - 3

print('Akarnya menggunakan Newton-Raphson =
     '+str(newton_raphson(persamaan,turunan,1)))
```

Hasil

```
>>>
======= RESTART: F:/TUGAS/Praktikum/Metnum/Tugas1/py/newton rapshon.py =======
Akarnya menggunakan Newton-Raphson =1.4462386859723786
>>> |
```

3.1.3 Penjelasan

Mencari f(xc) dan df(xc) terlebih dahulu, kemudian dilanjutkan xc menjadi $x_{c+1} = x_c - \frac{f(x_c)}{df(x_c)}$ dan mencari delta dari absolut $x_{c+1} - x_c$. Langkah terus dilakukan hingga kondisi berhenti yaitu delta ≤ eps & iter > maxit. Jika berhenti, didapatkan akar pendekatan yaitu x_{c+1}.

Akar pendekatannya : 1.4462

Dengan Toleransi : 2.81e-6

BAB IV
PENUTUPAN

4.1 Kesimpulan

Akar yang terdapat dari suatu persamaan nonlinear dapat dicari dengan Metode *Bisection* (Biseksi) dan Metode Newton-Raphson. Langkah-langkah yang tersedia dapat dibuat dengan menggunakan Bahasa *Python* atau menggunakan Spreadsheet seperti excel sehingga tidak perlu melakukan iterasi secara sistem manual seperti yang ada pada *proses* di atas. Hasil kerja dari sistem program tersebut adalah sebagai berikut.

1. Pada soal pertama tentang implementasi Metode *Bisection*, dicari akar dari persamaan $f(x) = e^x + x^2 - 3x - 2$ dengan batas kiri -1 dan batas kanan 1. Hasil yang diperoleh menggunakan metode biseksi yang telah dibuat dalam Bahasa *Python* adalah -0.39026641845703125. Excel juga menghasilkan nilai yang sama namun ditampilkan dalam pembulatan.

2. Pada soal kedua tentang implementasi Metode Newton-Raphson, dicari akar dari persamaan $f(x) = e^x + x^2 - 3x - 2 = 0$ yang memiliki turunan $f'(x) = e^x + 2x - 3$ dan harga taksiran 1. Hasil yang diperoleh menggunakan Metode Newton-Raphson yang telah dibuat dalam Bahasa *Python* adalah 1.4462386859723786. Excel juga menghasilkan nilai yang sama namun ditampilkan dalam pembulatan.