

LAPORAN

PRAKTIKUM KOMPUTASI BIOMEDIS

Chapter 2 : Finding-Root Method using Bisection and Regula-Falsi

Pelaksanaan Praktikum:

Hari: Senin

Tanggal: 12 Agustus 2019

Jam ke: 9-10



Oleh:

Nama : M. Thoriqul Aziz E

NIM : 081711733002

Dosen Pembimbing : Endah Purwanti, S. Si, M. T.

LABORATORIUM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS AIRLANGGA
SURABAYA

2019

A. TUJUAN

Mahasiswa dapat mengetahui akar dari persamaan menggunakan metode Biseksi dan Regula-Falsi.

B. DASAR TEORI

Metode numerik merupakan salah satu metode yang digunakan untuk menyelesaikan persoalan matematis. Metode ini diketahui lebih efisien dan efektif terlebih dengan dibantu oleh perangkat lunak pada komputer. Disisi lain metode numerik juga memberikan kajian parametric pada sebuah persoalan dengan batas medan yang bersifat sembarang. (Ismuniyanto, 2016)

Persamaan non-linier adalah persamaan untuk mencari nilai akar x sehingga $f(x) = 0$ yang dalam metode numerik, pencarian akan dilakukan dengan cara lelaran (iteratif). Secara umum metode yang digunakan adalah metode tertutup dan metode terbuka. Metode tertutup yaitu metode yang menggunakan dua nilai akar tebakan yang mengapit nilai akar fungsi. Sedangkan metode terbuka yaitu menebak nilai akar fungsi dengan satu nilai akar tebakan bebas.

Pada metode tertutup terdapat 2 cara yaitu metode Biseksi dan metode Regula-Falsi. Metode Biseksi menggunakan 2 akar tebakan yang mengapit akar dengan syarat memiliki tanda berbeda. Tebakan akar berikutnya ditentukan pada titik tengah selang bagian tempat terjadinya perubahan tanda. Dalam lelarannya, batas atas dan batas bawah yang mengapit memberikan harga bagi fungsi $f(x)$ untuk nilai $x =$ batas atas dan $x =$ batas bawah. Kemudian batas baru diperoleh dari nilai yang diberikan oleh fungsi hasil masukan batas bawah atau batas atas. Jika nilai fungsi batas tebakan berlainan dengan nilai fungsi batas bawah, maka batas lelaran baru pada metode biseksi adalah batas tebakan sebagai batas atas dan batas bawah tetap, begitu pula berlaku sebaliknya. Nilai lelaran bisa dianggap selesai jika nilai eror atau nilai fungsi pada batas tebakan akhir lebih kecil dari nilai toleransi yang diberikan.

Metode Regula-Falsi merupakan metode yang memanfaatkan titik titik hasil fungsi pada batas atas dan batas bawah dengan sebuah garis lurus. Perpotongan garis buatan dengan sumbu x merupakan tebakan baru dari nilai akar yang akan menjadi batas baru. Dalam pergeseran batas batas pengapit sama dengan

metode Biseksi. Jika dimisalkan batas tebakan akar adalah r dengan batas bawah adalah a dan batas atas adalah b , maka menentukan nilai r :

$$r = a - \frac{f(a)(a - b)}{f(a) - f(b)}$$

C. TUGAS

- 1) Find the root of equation system in following problem!

Packed bed column. A column packed with spherical particles provides high surface area geometry that is useful in isolating specific protein(s) or other biological molecules from a cell lysate mixture. The Ergun equation relates the pressure drop through a packed bed of spheres to various fluid and geometric parameters of the bed:

$$\frac{\Delta p}{l} = 150 \frac{(1 - \varepsilon)^2 \mu u}{d_p^2} + 1.75 \frac{(1 - \varepsilon) \rho u^2}{d_p}$$

Where Δp is the pressure drop, l is the length of the column, ε is the porosity, μ is the fluid viscosity, u is the fluid velocity, d_p is the diameter of spherical particles, and ρ is the density. for a 20 cm column, packed with 1 mm spheres and perfused with a buffer of equal viscosity and density to water($\mu = 0.01$ P, $\rho = 1\text{g/cm}^3$). By using the Bisection and Regula-Falsi method, determine the column porosity if the pressure drop for a fluid flowing is 810.5 dyn/cm² with velocity $u = 0.75$ cm/s. Make sure that you use consistent units throughout your calculation.

Using a starting interval of $0.1 < \varepsilon < 0.9$, report the number of iterations necessary to determine the porosity to within 0.01 (tolerance).

- 2) Analyze the advantages and disadvantages of both methods!

D. PEMBAHASAN

1. Permasalahan yaitu menyelesaikan persamaan Ergun. Sebelum menyelesaikan persamaan tersebut dengan metode komputasi, maka persamaan harus dibentuk terlebih dahulu menjadi persamaan kuadrat. Kemudian akan masuk dalam metode Biseksi dan Regula-Falsi. Berikut pengerjaan persamaan ergun dengan nilai nilai yang diketahui :

Diketahui :

$$l = 20\text{cm} ; d_p = 0.1\text{cm} ; \mu = 0.01 \text{ P} ; \rho = \frac{1g}{\text{cm}^3};$$

$$\Delta p = 810.5 \frac{\text{dyn}}{\text{cm}^2}; u = \frac{0.75\text{cm}}{\text{s}};$$

Dimisalkan:

$$\varepsilon = x ; a = (1 - x)$$

Maka :

$$\frac{\Delta p}{l} = \frac{150a^2\mu u + 1.75apu^2d_p}{d_p^2}$$

$$\frac{810.5 \times 0.1^2}{20} = 150 \times 0.01 \times 0.75(1 - 2x + x^2) + 1.75 \times 1 \times 0.75^2(1 - x)$$

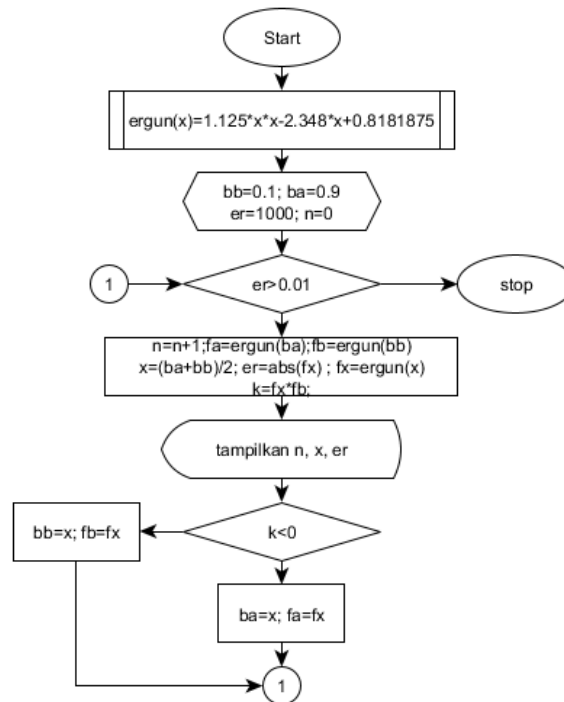
$$0 = 1.125x^2 - 2.348x + 0.8181875$$

Sehingga, persamaan menjadi :

$$y = 1.125x^2 - 2.348x + 0.8181875$$

A. Metode Biseksi

Setelah mengetahui penyederhanaan persamaan ergun, maka menggunakan metodek Biseksi dengan batas batas nilai x yang ditentukan. Dari batas tersebut kemudian melakukan looping dalam menebak nilai x yang benar. Berikut adalah diagram flowchart program:



Berikut kode program dalam IDE Octave 5.10:

```
4 function ap=ergun(x)
5 ap=1.125*x*x-2.348*x+0.8181875;
6 endfunction
7 bb=0.1;
8 ba=0.9;
9 er=100000;
10 n=0;
11 fprintf("Iterasi      x      error\n");
12 while er>0.01
13     n=n+1;
14     fa=ergun(ba);
15     fb=ergun(bb);
16     x=(ba+bb)/2;
17     fx=ergun(x);
18     er=abs(fx);
19     k=fx*fb;
20     fprintf("%d      %d      %f\n",n,x,er);
21     if k<0
22         ba=x;
23         fa=fx;
24     else
25         bb=x;
26         fb=fx;
27     endif
28 endwhile
```

Pada syntax:

```
function ap=ergun(x)
ap=1.125*x*x-2.348*x+0.8181875;
endfunction
```

menunjukkan fungsi ergun yang telah di sederhanakan sebelumnya. Kemudian dibentuk dalam syntax function untuk memanggil persamaan fungsi tanpa menuliskan kembali fungsi secara utuh.

Pada syntax:

```
bb=0.1;
ba=0.9;
er=100000;
n=0;
fprintf("Iterasi      x      error\n");
```

variable bb menunjukkan nilai batas bawah dan variable ba untuk batas atas dalam metode Biseksi. Nilai variable er digunakan untuk mengenalkan nilai eror mula mula. Dan variable n untuk menunjukkan

jumlah iterasi yang nanti akan dilakukan oleh system. Syntax `fprintf` sebagai pembuatan format table(kolom atas) pada *command window*.

Pada syntax:

```
while er>0.01
```

menunjukkan jenis perulangan yaitu tipe perulangan bersyarat dengan batas tak pasti. Syntax `while` akan berhenti jika nilai eror pada variable `er` akan menunjukkan angka lebih dari 0,01.

Pada syntax:

```
n=n+1;
```

```
fa=ergun(ba);
```

```
fb=ergun(bb);
```

```
x=(ba+bb)/2;
```

syntax `n=n+1;` menunjukkan bahwa nilai iterasi akan terus bertambah 1 sesuai dengan jumlah perulangan yang terjadi. Kemudian syntax `fa=ergun(ba);` menunjukkan hasil persamaan fungsi `ergun` dengan nilai `x=ba` yang kemudian akan didefinisikan ulang menjadi variable `fa`, begitu pula dengan syntax `fb=ergun(bb);` mendefinisikan ulang nilai fungsi `ergun` pada nilai `x=bb` pada variable `fb`. Syntax `x=(ba+bb)/2;` menunjukkan nilai tengah yang kemudian nanti akan menjadi batas baru dari metode Biseksi.

Pada syntax:

```
fx=ergun(x);
```

```
er=abs(fx);
```

```
k=fx*fb;
```

```
fprintf("%d          %d          %f\n",n,x,er);
```

syntax `fx=ergun(x);` menunjukkan fungsi `ergun` dalam nilai `x` yang kemudian akan didefinisikan dalam variable `fx`. Variable `er` menunjukkan nilai eror yang didapatkan dengan memasukan nilai `x` dalam fungsi, yang kemudian nilainya di-absolute-kan. Syntax `k=fx*fb;` menunjukkan nilai perkalian `fx` dan `fb` sebagai penentu seleksi perubahan batas atas atau batas bawah. Syntax `fprintf` menunjukkan hasil nilai dari variable yaitu `n`, `x` dan `er`.

Pada syntax:

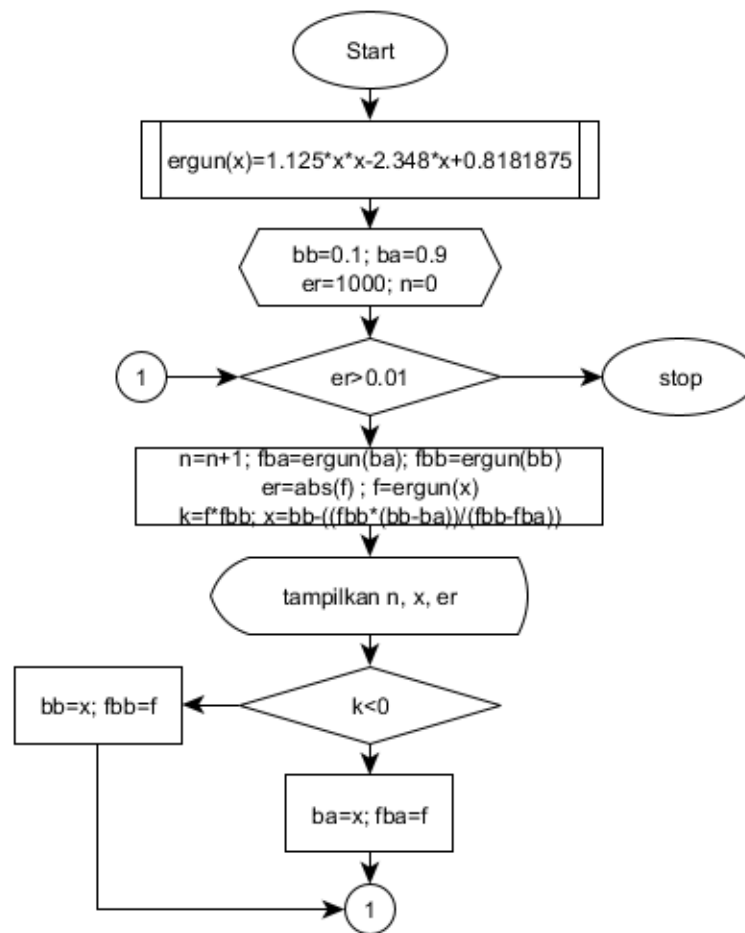
```
if k<0
    ba=x;
    fa=fx;
else
    bb=x;
    fb=x;
endif
```

program seleksi ini untuk menentukan perubahan batas atas atau bawah menurut metode Biseksi dimana terdapat syarat yaitu nilai fungsi pada batas bawah dan batas atas harus berlainan tanda. If $k < 0$, menunjukkan bahwa jika nilai k sebelumnya bernilai kurang dari 0 (negatif) maka statement yang akan dilakukan adalah merubah batas atas ba menjadi x dan nilai fungsi fa menjadi fx . Hal yang sama juga terjadi sebaliknya pada variable bb dan fb jika seleksi bernilai salah. Berikut adalah hasil pada *command window*:

Command Window		
Iterasi	x	error
1	0.5	0.074562
2	0.3	0.215037
3	0.4	0.058987
4	0.45	0.010600
5	0.425	0.023491
6	0.4375	0.006270
>>		

B. Metode Regula-Falsi

Setelah mengetahui penyederhanaan persamaan ergun, maka menggunakan metodek Biseksi dengan batas batas nilai x yang ditentukan. Dari batas tersebut kemudian melakukan looping dalam menebak nilai x yang benar. Berikut adalah diagram flowchart program:



Berikut kode program dalam IDE Octave 5.10:

```

4 function ap=ergun(x)
5 ap=1.125*x*x-2.348*x+0.8181875;
6 endfunction
7 bb=0.1;
8 ba=0.9;
9 er=1000;
10 n=0;
11 fprintf("iteraasi   x           fba       fbb       fx       error\n");
12 while er>0.01
13     n=n+1;
14     fba=ergun(ba);
15     fbb=ergun(bb);
16     x=bb-((fbb*(bb-ba))/(fbb-fba));
17     f=ergun(x);
18     k=f*fbb;
19     er=abs(f);
20     fprintf("%d      %d,      %d,      %d,      %d,      %d\n",n,x,fba,fbb,f,er);
21     if k<0
22         ba=x;
23         fba=f;
24     else
25         bb=x;
26         fbb=f;
27     endif
28 endwhile
  
```


Pada syntax:

```
function ap=ergun(x)
ap=1.125*x*x-2.348*x+0.8181875;
endfunction
```

menunjukkan fungsi `ergun` yang telah di sederhanakan sebelumnya. Kemudian dibentuk dalam syntax `function` untuk memanggil persamaan fungsi tanpa menuliskan kembali fungsi secara utuh.

Pada syntax:

```
bb=0.1;
ba=0.9;
er=1000;
n=0;
fprintf("iterasi x   fba fbb  fx      eror\n");
```

variable `bb` menunjukkan nilai batas bawah dan variable `ba` untuk batas atas dalam metode Regula-Falsi. Nilai variable `er` digunakan untuk mengenalkan nilai eror mula mula. Dan variable `n` untuk menunjukan jumlah iterasi yang nanti akan dilakukan oleh system. Syntax `fprintf` sebagai pembuatan format table(kolom atas) pada *command window*.

Pada syntax:

```
while er>0.01
```

menunjukkan jenis perulangan yaitu tipe perulangan bersyarat dengan batas tak pasti. Syntax `while` akan berhenti jika nilai eror pada variable `er` akan menunjukan angka lebih dari 0,01.

Pada syntax:

```
n=n+1;
fba=ergun(ba);
fbb=ergun(bb);
x=bb-((fbb*(bb-ba))/(fbb-fba));
```

syntax `n=n+1;` menunjukan bahwa nilai iterasi akan terus bertambah 1 sesuai dengan jumlah perulangan yang terjadi. Kemudian syntax `fba=ergun(ba);` menunjukan hasil persamaan fungsi `ergun` dengan nilai `x=ba` yang kemudian akan didefinisikan ulang menjadi variable

fba, begitu pula dengan syntax `fbf=ergun(bb);` mendefinisikan ulang nilai fungsi ergun pada nilai `x=bb` pada variable `fbf`. Syntax `x=bb-((fbf*(bb-ba))/(fbf-fba));` menunjukan nilai tebakan `x` berdasarkan perpotongan garis yang kemudian nanti akan menjadi batas baru dari metode Regula-Falsi.

Pada syntax:

```
f=ergun(x);
k=fbf*f;
er=abs(f);
fprintf("%d%d,      %d,      %d,      %d,
%d\n",n,x,fba,fbf,f,er);
```

syntax `f=ergun(x);` menunjukan fungsi ergun dalam nilai `x` yang kemudian akan didefinisikan dalam variable `f`. Variable `er` menunjukan nilai eror yang didapatkan dengan memasukan nilai `x` dalam fungsi, yang kemudian nilainya di-absolute-kan. Syntax `k=fbf*f;` menunjukan nilai perkalian `f` dan `fbf` sebagai penentu seleksi perubahan batas atas atau batas bawah. Syntax `fprintf` menunjukan hasil nilai dari variable yaitu `n, x, fba, fbf, f` dan `er`.

Pada syntax:

```
if k<0
    ba=x;
    fa=fx;
else
    bb=x;
    fb=x;
endif
```

program seleksi ini untuk menentukan perubahan batas atas atau bawah menurut metode Regula-Falsi dimana terdapat syarat yaitu nilai fungsi pada batas bawah dan batas atas harus berlainan tanda. If `k<0` , menunjukkan bahwa jika nilai `k` sebelumnya bernilai kurang dari 0 (negatif) maka statement yang akan dilakukan adalah merubah batas atas

ba menjadi x dan nilai fungsi fba menjadi f. Hal yang sama juga terjadi sebaliknya pada variable bb dan fbb jika seleksi bernilai salah. Berikut adalah hasil pada *command window*:

Command Window					
iteraasi	x	fba	fbb	fx	eror
1	0.586212,	-0.383763,	0.594637,	-0.171638,	0.171638
2	0.477305,	-0.171638,	0.594637,	-0.0462275,	0.0462275
3	0.450089,	-0.0462275,	0.594637,	-0.0107191,	0.0107191
4	0.44389,	-0.0107191,	0.594637,	-0.00239827,	0.00239827
>>					

- Untuk kelebihan dan kekurangan dua metode diatas adalah yang pertama untuk metode Biseksi, kelebihanannya adalah secara matematis lebih mudah diaplikasikan karena rumus penghitungan yang tidak lebih rumit dari metode Regula-Falsi. Sedangkan kekurangannya adalah cenderung membutuhkan iterasi yang lebih banyak untuk menemukan nilai akar dengan toleransi yang sama dengan metode Regula-Falsi.

Sedangkan kelebihan dari metode Regula-Falsi, iterasi yang dibutuhkan untuk mencari nilai akar cenderung lebih sedikit dibanding metode Biseksi. Sedangkan secara matematis, metode Regula-Falsi lebih rumit dibanding dengan Biseksi.

E. KESIMPULAN

. Dari hasil pengolahan data pada Octave 5.10 didapatkan nilai akar pada metode Biseksi adalah 0.4375 dengan iterasi sebanyak 6 kali. Sedangkan pada metode Regula-Falsi didapatkan nilai akar adalah 0.44389 dengan jumlah iterasi sebanyak 4 kali.

F. DAFTAR PUSTAKA

Capra, Steven C and Canale.1991. “**Numerical Methods for Engineers with Personal Computers Applications**”. MacGraw-Hill Book Company.

Ismuniyarto.2016.” **Perbandingan Metode Pengapitan Akar (*Bisection, Regula Falsi dan Secant*) Persamaan Non Linear dalam Menyelesaikan Analisis Break Even**”. Universitas Islam Negeri Alauddin Makassar. Diakses pada 25 Agustus 2019.

King M.R and Mody N.A .2010. “**Numerical and Statical Methods for Bioengineering**”.Cambridge University Press. New York

