**PERTEMUAN 9**

**TEKNIK PENCARIAN AKAR PADA PERSAMAAN NON-LINIER**



**TUJUAN PRAKTIKUM**

Mahasiswa mampu menerapkan teknik-teknik pencarian akar pada persamaan non-linier menggunakan Program R.



**TEORI PENUNJANG**

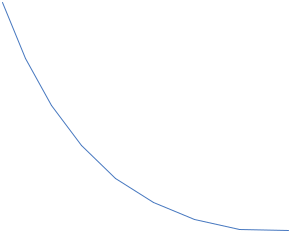
**Metode Bagi Dua**

Misalkan *f*(*x*) kontinu pada selang [*a, b*] dimana *f*(*a*).*f*(*b*)  0. Hal ini menunjukkan bahwa *f*(*x*) berubah tanda pada [*a, b*] yang artinya *f*(*x*) = 0 mempunyai sedikitnya 1 akar pada [*a, b*] (lihat Gambar 1).

*y*



*f(x)*



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| *a* | | *c* | | *b* | *x* |



Gambar 1. Grafik fungsi *f*(*x*) pada selang

Proses pencarian akar pada *f*(*x*) = 0 pada selang [*a, b*] dapat dilakukan dengan membagi selang tersebut menjadi dua bagian, yaitu [*a*, *c*] dan [*c*, b], sehingga berlaku

Selanjutnya untuk mencari akar *f*(*x*) = 0 pada selang [*a*, *c*] dapat dilakukan dengan membagi selang tersebut menjadi dua bagian, yaitu [*a*, *c*1 ] dan [*c*1, *c*], sehingga berlaku

Demikian proses penentuan akar tersebut berlanjut, sehingga jika akar terdapat pada selang [a,*c*1], maka selang tersebut dibagi dua sehingga diperoleh selang-selang [*a*, *c*2 ] dan [*c*2, *c*1]

*a**c*1

dengan *c*2 = 2 . Proses tersebut akan berhenti bila

1. Akar telah ditemukan
2. Mencapai iterasi maksimum (N) yang telah ditetapkan sebelumnya
3. *b*  *c* (lebar selang cukup kecil).

Berikut ini adalah algoritma pencarian akar dari *f*(*x*) = 0 dengan menggunakan metode bagi dua.

1. Input : *a*, *b*,  dengan *f*(*a*).*f*(*b*)  0.

2. Set

jika *f*(*c*) = 0 maka akar = *c*  proses selesai

jika *f*(*a*).*f*(*c*)  0 maka set *b* = *c*

jika tidak, set *a* = *c*

{periksa apakah *b*  *a*}

jika *b*  a proses selesai  akar = *c*

jika tidak, kembali ke langkah 2.

**Contoh 1:**

Akar pada selang [0, 0.5] dengan dapat ditentukan sebagai berikut (perhitungan menggunakan decimal 4 digit). Berikut adalah nilai *a, b, c* dan hasil perhitungan dalam Contoh 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabel 1 | Metode Bagi Dua untuk pada selang [0, 0.5] | | | | |
| iterasi | *a* | *b* | *c* | *f* (*c* ) |  |
| 1 | 0.0000 | 0.5000 | 0.2500 | 0.5156 |  |
|  |  |  |  |  |  |
| 2 | 0.0000 | 0.2500 | 0.1250 | -0.2480 |  |
|  |  |  |  |  |  |
| 3 | 0.1250 | 0.2500 | 0.1875 | 0.1315 |  |
|  |  |  |  |  |  |
| 4 | 0.1250 | 0.1875 | 0.1562 | -0.05868 |  |
|  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |  |  |  |
| 5 | 0.1562 | 0.1875 | 0.1718 |  | 0.03632 |  |
|  |  |  |  |  |  |  |
| 6 | 0.1562 | 0.1718 | 0.1640 |  | -0.0112 |  |
|  |  |  |  |  |  |  |

Dari proses perhitungan dapat dilihat bahwa lebar selang pada setiap iterasi diberikan dalam tabel berikut.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| iterasi | 1 | 2 | 3 | 4 | 5 | 6 |
| *b*  *a* | 0.5 | 0.25 | 0.125 | 0.0625 | 0.0313 | 0.0156 |

Secara umum, jika nilai ujung selang pada iterasi ke-i dinotasikan sebagai *a*i dan *b*i, untuk

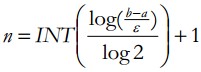
1. = 1, 2,...,*n*, maka lebar selang pada saat ke-*i* adalah

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | | (*b* *a*). | (1) | |  |  |  |
|  | 2i |  |
|  |  |  |  |  |  |  |
| Proses pencarian akar akan berhenti bila lebar selang sangat kecil, artinya | | |  | *bn*  *a n* |  |  ** . |  |
|  |  |  |

Untuk nilai  yang diketahui, kita dapat menentukan banyaknya iterasi sampai akar diperoleh. Pada iterasi ke-*n*, proses pencarian akar akan berhenti karena

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *bn* |  *a n* |  | ** | (2) |  |
|  |  |  |
| dengan menggunakan persamaan (1), persamaan (2) menjadi | | | | |  |  |
| 1 | | *b*  *a*  ** | | | (3) |  |
|  | 2 *n* |  |
|  |  |  |  |  |  |

Selanjutnya dari persamaan (3) diperoleh



**Contoh 2:**

Tentukan akar terbesar dari persamaan

*f*(*x*)= *e x* 4*x*

pada selang [0, 0.5] dengan  =0.01 dan lakukan iterasi hingga 20

**Penyelesaian:**

Tabel berikut menyatakan nilai *a*, *b*, *c* dan *f*(*c* ) untuk *f*(*x*)= *ex*  4*x* pada Contoh 2

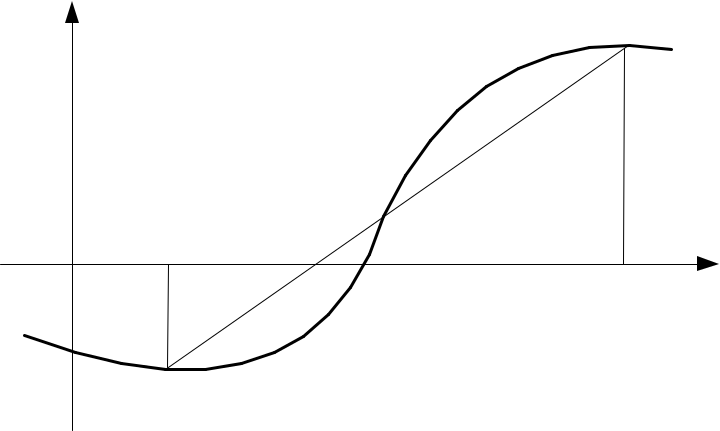
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| iterasi | *a* | *b* | *c* | *b - a* | *f*(*c* ) |
| 1 | 0.0000 | 0.5000 | 0.2500 | 0.5000 | 0.2840 |
|  |  |  |  |  |  |
| 2 | 0.2500 | 0.5000 | 0.3750 | 0.2500 | -0.0450 |
|  |  |  |  |  |  |
| 3 | 0.2500 | 0.3750 | 0.3125 | 0.1250 | 0.1168 |
|  |  |  |  |  |  |
| 4 | 0.3125 | 0.3750 | 0.3437 | 0.0625 | 0.0352 |
|  |  |  |  |  |  |
| 5 | 0.3437 | 0.3750 | 0.3593 | 0.0313 | -0.0050 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 6 | 0.3437 | 0.3593 | 0.3515 | 0.0156 | 0.0150 |
|  |  |  |  |  |  |

**Metode Regula-falsi**

Seperti yang pernah kita bahas sebelumnya, metode bagidua mempunyai kelemahan yaitu kecepatan konvergensinya sangat lambat. Kecepatan konvergensi dapat ditingkatkan dengan memperhitungkan nilai f(a) dan f(b). Logikanya, bila f(a) lebih dekat ke nol daripada f(b) tentu akar lebih dekat ke x = a daripada ke x = b. Metode yang memanfaatkan nilai f(a) dan f(b) ini adalah metode regula-falsi (bahasa Latin) atau metode posisi palsu (*false position method*). Berikut adalah gambar metode regula-falsi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *y* |  | *B* |  |  |
|  |  |  |  |
|  |  | *y* = *f*(*x*) |  |  |
| *a* | *C* |  |  |  |
|  | *c* | *b* | *x* |  |
| *A* |  |  |  |  |



Jika gradien garis AB = gradien garis BC, maka

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *f* *b* *f* *a* | |  | *f* (*b*)0 | |  |
|  | | *b*  *c* | |  |
| *b*  *a* |  |  |  |
| *c*  *b*  | *f* *b**b*  *a* | | | |  |
| *f* *b* *f* *a* | | |  |  |

Seperti dalam Metode Bagi Dua, terdapat 3 kemungkinan letak akar yaitu

* f(a) dan f(c ) memiliki tanda yang berbeda dan akar terletak dalam [a, c ], atau
* f(c ) dan f(b) memiliki tanda yang berbeda dan akar terletak dalam [c, b], atau
* f(c ) = 0 yang berarti akar ditemukan.

Ketiga kemungkinan di atas digunakan untuk menentukan barisan interval [an, bn].

Pendekatan ke akar pada setiap langkah ditentukan sebagai berikut :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *cn* |  *bn* |  | *f* *bn* *bn*  *an*  | | | |  |
| *f* *bn* |  *f* *an* |  |  |  |
|  |  |  |  |

Dapat ditunjukkan bahwa barisan {cn} akan konvergen ke akar r.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Contoh 2:**  Hitunglah akar | | |  | di dalam selang | [0, 0.5] | dan dengan |  |
| maksimum | | | iterasi adalah 3 |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | iterasi | *ak* | *ck* | *bk* | *f(ck)* |  |
|  |  | 1 | 0.0000000 | 0.3636364 | 0.5 | 4.282494e-02 | |
| 2 | | | 0.3636364 | 0.3686369 | 0.5 | 8.094649e-04 | |
| 3 | | | 0.3686369 | 0.3687314 | 0.5 | 1.534145e-05 | |
|  |  |  |  |  |  |  |  |



**LAPORAN PENDAHULUAN**

1. Jelaskan secara singkat bagaimana proses penentuan akar pada metode bagi dua dan Regula-Falsi!
2. Apa yang menyebabkan iterasi proses penentuan akar berhenti pada metode bagi dua dan Regula-Falsi?
3. Jelaskan secara singkat kelebihan dan kekurangan metode bagi dua dan Regula - Falsi!



**MATERI PRAKTIKUM**

1. Proses penentuan akar dengan metode bagi dua dan Regula-Falsi.
2. Buatlah program menggunakan bahasa R untuk menentukan nilai akar dari persamaan pada Contoh 1.
3. Diberikan fungsi pada Contoh 2.
   1. Dengan menggunakan R, hitunglah nilai a, b, c, dan f(c) untuk fungsi tersebut. Lakukan hingga iterasi 20.
   2. Buatlah Tabel nilai a, b, c dan f(c) yang telah dihitung pada poin a.
   3. Mengapa iterasi berhenti pada iterasi ke 6?
4. Buat pula program R untuk menyelesaikan permasalah pada Contoh 3.
5. Bandingkan hasil yang didapatkan dari point 3 dan 4.



**DAFTAR PUSTAKA**

1. Atkinson K. 1994. Elementary Numerical Analysis, Second edition. Wiley
2. Victor A. Bloomfield. 2014. *Using R for Numerical Analysis in Science and* *Engineering*. 1 edition. Chapman and Hall/CRC

1. [Steven C.](http://www.amazon.com/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=Steven+Chapra&search-alias=books&text=Steven+Chapra&sort=relevancerank) 2011. *Applied Numerical Methods W/MATLAB: for Engineers & Scientists*. 3 edition. McGraw-Hill