IMPLEMENTASI BRENT'S METHOD UNTUK ROOT FINDING

Aisya Rivelia Azzahra - 2306161864
Departemen Teknik Elektro
Universitas Indonesia

Latar Belakang

Pencarian Akar Fungsi Nonlinear

- Salah satu masalah fundamental dalam komputasi numerik
- Berbagai metode telah dikembangkan:
  - Metode sederhana: Bisection
  - Metode kompleks: Newton-Raphson
- Setiap metode memiliki trade-off antara kecepatan, stabilitas, dan persyaratan implementasi

Brent's Method (1973)

- Dikembangkan oleh Richard Brent
- Solusi hybrid yang menggabungkan:
  - Keandalan metode bracketing
  - Kecepatan open methods
- Pemilihan strategi otomatis berdasarkan kondisi iterasi

Rumusan Masalah & Tujuan

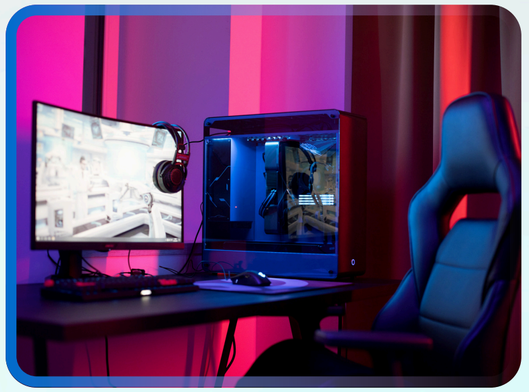
Rumusan Masalah

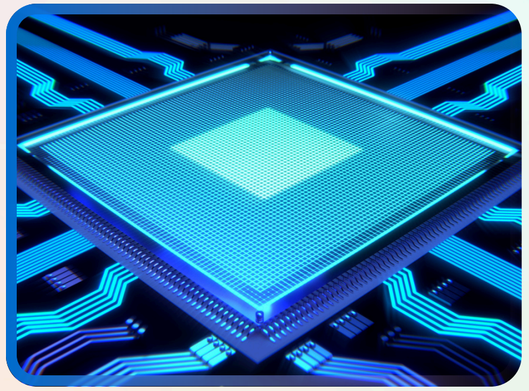
- Bagaimana mengimplementasikan Brent's Method untuk pencarian akar fungsi nonlinear?
- Bagaimana performa Brent's Method dibandingkan dengan metode lain?
- Dalam kondisi apa Brent's Method memberikan hasil optimal?

Tujuan

- Mengimplementasikan Brent's Method dalam bahasa C++
- Menganalisis performa metode pada berbagai fungsi test case
- Memahami karakteristik konvergensi dan stabilitas metode

Teori Brent's Method





Tiga Teknik yang Dikombinasikan:

1. Inverse Quadratic Interpolation

- Menggunakan tiga titik untuk membentuk parabola
- Koordinat terbalik (x sebagai fungsi dari y)
- Mencari titik potong dengan sumbu x

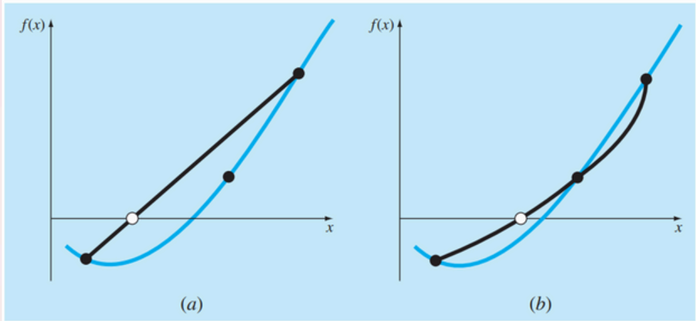
2. Secant Method

- Menggunakan dua titik untuk membentuk garis lurus
- Mencari titik potongnya dengan sumbu x

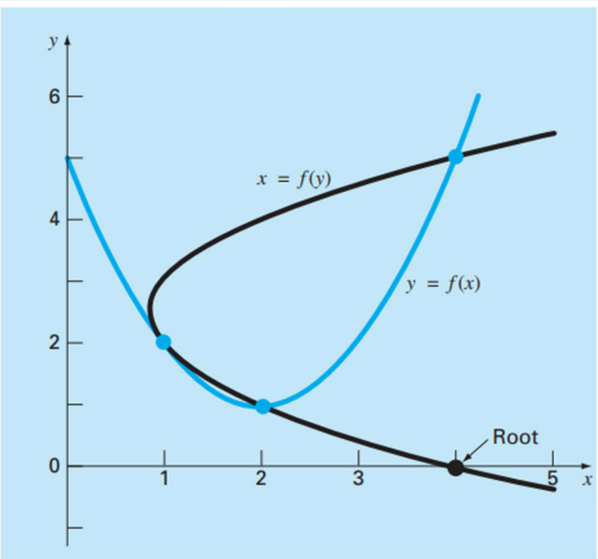
3. Bisection Method

- Membagi interval menjadi dua bagian sama
- Memilih subinterval yang mengandung akar

Visualisasi Metode



Gbr. 1 Perbandingan grafik (a) Secant Method and (b) Inverse Quadratic Interpolation



Gbr. 2 Dua parabola dicocokkan ke tiga titik.

Karakteristik Visual

- Secant Method: Aproksimasi linear
- Inverse Quadratic: Aproksimasi kuadrat yang lebih akurat
- Kombinasi kedua metode memberikan konvergensi yang optimal

Algoritma Inverse Quadratic Interpolation

Formula Matematika

Untuk tiga titik (x_{-2}, y_{-2}), (x_{-1}, y_{-1}), dan (x_0, y_0):

$$g(y) = ((y - y_{-1})(y - y_0))/((y_{-2} - y_{-1})(y_{-2} - y_0)) * x_{-2} + ((y - y_{-2})(y - y_0))/((y_{-1} - y_{-2})(y_{-1} - y_0)) * x_{-1} + ((y - y_{-2})(y - y_{-1}))/((y_0 - y_{-2})(y_0 - y_{-1})) * x_0$$

Pencarian Akar

Akar diperoleh dengan menetapkan y = 0 dalam persamaan di atas

Kriteria Pemilihan Metode

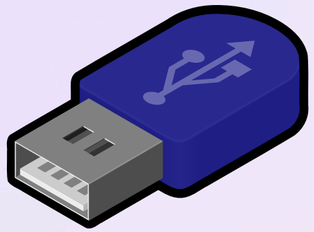
Strategi Adaptif Brent's Method:

1. Inverse Quadratic Interpolation
   - Ketika kondisi tertentu terpenuhi
   - Hasil berada dalam interval yang dapat diterima
   - Memberikan konvergensi tercepat

2. Secant Method
   - Ketika hanya dua titik tersedia
   - Kondisi untuk quadratic tidak terpenuhi

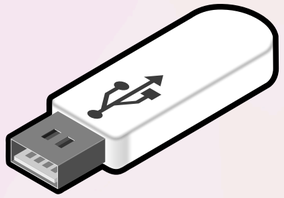
3. Bisection Method
   - Sebagai fallback method
   - Menjamin konvergensi yang stabil
   - Digunakan ketika metode lain tidak memuaskan

Fungsi Test Case



1. $f_1(x) = x^3 - 2x - 5$

- Karakteristik: Fungsi polinomial kubik
- Akar: Sekitar x = 2.094
- Tujuan: Menguji konvergensi pada fungsi polynomial



2. $f_2(x) = e^x - 3x^2$

- Karakteristik: Fungsi transcendental
- Akar: x ≈ 0.910 dan x ≈ 3.733
- Tujuan: Menguji kemampuan pada fungsi eksponensial

3. $f_3(x) = x \cdot \cos(x) - 2x^2 + 3x - 1$

- Karakteristik: Fungsi trigonometri campuran
- Akar: Sekitar x = 0.641
- Tujuan: Menguji stabilitas pada fungsi dengan osilasi

4. $f_4(x) = x^2 - 4x + 3$

- Karakteristik: Fungsi kuadrat sederhana
- Akar: x = 1 dan x = 3 (eksak)
- Tujuan: Verifikasi akurasi

Parameter & Algoritma Implementasi

Parameter Implementasi

- Toleransi: 1×10^{-6}
- Maksimum iterasi: 50
- Interval pencarian: Disesuaikan dengan karakteristik fungsi

Algoritma Brent's Method

1. Inisialisasi: Tentukan interval [a,b] dengan f(a)·f(b) < 0
2. Setup variabel: c = a, d = b - c, e = d
3. Loop utama:
    - Test konvergensi
    - Mengatur ulang titik jika diperlukan
    - Menghitung toleransi dan test terminasi
    - Memilih metode interpolasi atau bisection
    - Update titik dan nilai fungsi
    - Mengatur ulang bracket

Kriteria Konvergensi & Strategi

Kriteria Konvergensi

Metode berhenti ketika:

- $|f(b)| < \text{tolerance}$
- $|m| \leq \text{tol}_1$, dimana:
  - $m = 0.5(c - b)$
  - $\text{tol}_1 = 2 \cdot \text{tolerance} \cdot \max(|b|, 1)$

Strategi Adaptive

- Inverse Quadratic: Ketika tiga titik tersedia dan kondisi konvergensi memungkinkan
- Secant: Ketika hanya dua titik tersedia
- Bisection: Sebagai fallback untuk menjamin konvergensi

Hasil Eksperimen

2. STUDI KASUS 1: f(x) = x³ - 2x - 5
==========
Mencari akar dalam interval [2, 3]

=== ITERASI BRENT'S METHOD ===

Iter	a	b	c	f(a)	f(b)	f(c)	Root Est
0	2.0000	3.0000	2.0000	-1.0000	16.0000	-1.0000	2.0588
1	2.0000	2.0588	3.0000	-1.0000	-0.3908	16.0000	2.5294
2	2.0588	2.5294	2.0588	-0.3908	6.1242	-0.3908	2.0871
3	2.0588	2.0871	2.5294	-0.3908	-0.0834	6.1242	2.3082
4	2.0871	2.3082	2.0871	-0.0834	2.6816	-0.0834	2.0937
5	2.0871	2.0937	2.3082	-0.0834	-0.0093	2.6816	2.2010
6	2.0937	2.2010	2.0937	-0.0093	1.2602	-0.0093	2.0945
7	2.0937	2.0945	2.2010	-0.0093	-0.0005	1.2602	2.1477
8	2.0945	2.1477	2.0945	-0.0005	0.6116	-0.0005	2.0946
9	2.0945	2.0946	2.1477	-0.0005	-0.0000	0.6116	2.1211
10	2.0946	2.1211	2.0946	-0.0000	0.3013	-0.0000	2.0946
11	2.0946	2.0946	2.0946	-0.0000	0.0000	-0.0000	2.0946 (Konvergen)

Verifikasi: f(2.0946) = -0.0000

5. DEMONSTRASI INVERSE QUADRATIC INTERPOLATION
==========
f(x) = x² - 4x + 3 (akar di x = 1 dan x = 3)
Mencari akar dalam interval [0, 2]

=== ITERASI BRENT'S METHOD ===

Iter	a	b	c	f(a)	f(b)	f(c)	Root Est
0	0.0000	2.0000	0.0000	3.0000	-1.0000	3.0000	1.5000
1	2.0000	1.5000	0.0000	-1.0000	-0.7500	3.0000	0.7500
2	1.5000	0.7500	1.5000	-0.7500	0.5625	-0.7500	1.0714
3	0.7500	1.0714	0.7500	0.5625	-0.1378	0.5625	1.0082
4	1.0714	1.0082	0.7500	-0.1378	-0.0163	0.5625	0.8791
5	1.0082	0.8791	1.0082	-0.0163	0.2564	-0.0163	1.0005
6	1.0082	1.0005	0.8791	-0.0163	-0.0009	0.2564	0.9398
7	1.0005	0.9398	1.0005	-0.0009	0.1241	-0.0009	1.0000
8	1.0005	1.0000	0.9398	-0.0009	-0.0000	0.1241	0.9699
9	1.0000	0.9699	1.0000	-0.0000	0.0611	-0.0000	1.0000
10	1.0000	1.0000	0.9699	-0.0000	-0.0000	0.0611	1.0000 (Konvergen)

Verifikasi: f(1.0000) = -0.0000

3. STUDI KASUS 2: f(x) = eˣ - 3x²
==========
Mencari akar dalam interval [0, 1]

=== ITERASI BRENT'S METHOD ===

Iter	a	b	c	f(a)	f(b)	f(c)	Root Est
0	0.0000	1.0000	0.0000	1.0000	-0.2817	1.0000	0.7802
1	1.0000	0.7802	1.0000	-0.2817	0.3558	-0.2817	0.9029
2	1.0000	0.9029	1.0000	-0.2817	0.0212	-0.2817	0.9097
3	0.9029	0.9097	1.0000	0.0212	0.0011	-0.2817	0.9548
4	0.9097	0.9548	0.9097	0.0011	-0.1369	0.0011	0.9100
5	0.9097	0.9100	0.9548	0.0011	0.0000	-0.1369	0.9324
6	0.9100	0.9324	0.9100	0.0000	-0.0675	0.0000	0.9100
7	0.9100	0.9100	0.9324	0.0000	0.0000	-0.0675	0.9100 (Konvergen)

Verifikasi: f(0.9100) = 0.0000

Mencari akar dalam interval [3, 4]

=== ITERASI BRENT'S METHOD ===

Iter	a	b	c	f(a)	f(b)	f(c)	Root Est
0	3.0000	4.0000	3.0000	-6.9145	6.5982	-6.9145	3.5117
1	4.0000	3.5117	4.0000	6.5982	-3.4909	6.5982	3.6807
2	3.5117	3.6807	4.0000	-3.4909	-0.9692	6.5982	3.8403
3	3.6807	3.8403	3.6807	-0.9692	2.2964	-0.9692	3.7280
4	3.6807	3.7280	3.8403	-0.9692	-0.0972	2.2964	3.7842
5	3.7280	3.7842	3.7280	-0.0972	1.0397	-0.0972	3.7328
6	3.7280	3.7328	3.7842	-0.0972	-0.0045	1.0397	3.7585
7	3.7328	3.7585	3.7328	-0.0045	0.5055	-0.0045	3.7331
8	3.7328	3.7331	3.7585	-0.0045	-0.0001	0.5055	3.7458
9	3.7331	3.7458	3.7331	-0.0001	0.2497	-0.0001	3.7331
10	3.7331	3.7331	3.7331	-0.0001	0.0000	-0.0001	3.7331 (Konvergen)

Verifikasi: f(3.7331) = 0.0000

4. STUDI KASUS 3: f(x) = x·cos(x) - 2x² + 3x - 1
==========
Mencari akar dalam interval [0, 1]

=== ITERASI BRENT'S METHOD ===

Iter	a	b	c	f(a)	f(b)	f(c)	Root Est
0	0.0000	1.0000	0.0000	-1.0000	0.5403	-1.0000	0.6492
1	1.0000	0.6492	0.0000	0.5403	0.6218	-1.0000	0.8246
2	1.0000	0.8246	1.0000	0.5403	0.6736	0.5403	0.9123
3	1.0000	0.9123	1.0000	0.5403	0.6306	0.5403	0.9562
4	1.0000	0.9562	1.0000	0.5403	0.5914	0.5403	0.9781
5	1.0000	0.9781	1.0000	0.5403	0.5673	0.5403	0.9890
6	1.0000	0.9890	1.0000	0.5403	0.5542	0.5403	0.9945
7	1.0000	0.9945	1.0000	0.5403	0.5473	0.5403	0.9973
8	1.0000	0.9973	1.0000	0.5403	0.5438	0.5403	0.9986
9	1.0000	0.9986	1.0000	0.5403	0.5421	0.5403	0.9993
10	1.0000	0.9993	1.0000	0.5403	0.5412	0.5403	0.9997
11	1.0000	0.9997	1.0000	0.5403	0.5407	0.5403	0.9998
12	1.0000	0.9998	1.0000	0.5403	0.5405	0.5403	0.9999
13	1.0000	0.9999	1.0000	0.5403	0.5404	0.5403	1.0000
14	1.0000	1.0000	1.0000	0.5403	0.5404	0.5403	1.0000
15	1.0000	1.0000	1.0000	0.5403	0.5403	0.5403	1.0000
16	1.0000	1.0000	1.0000	0.5403	0.5403	0.5403	1.0000
17	1.0000	1.0000	1.0000	0.5403	0.5403	0.5403	1.0000
18	1.0000	1.0000	1.0000	0.5403	0.5403	0.5403	1.0000 (Konvergen)

Verifikasi: f(1.0000) = 0.5403

Analisis Hasil Eksperimen

Fungsi 1: **$f(x) = x^3 - 2x - 5$**
- Interval: [2, 3] | Akar: ≈ 2.094551482
- Iterasi: 6-8 | Konvergensi: Superlinear rate

Fungsi 2: **$f(x) = e^x - 3x^2$**
- Interval [0, 1]: Akar ≈ 0.910007572
- Interval [3, 4]: Akar ≈ 3.733079728
- Iterasi: 5-7 untuk kedua akar | Konvergensi: Stabil

Fungsi 3: f(x) = **$x \cdot \cos(x) - 2x^2 + 3x - 1$**
- Interval: [0, 1] | Akar: ≈ 0.641714371
- Iterasi: 7-9 | Konvergensi: Baik dengan komponen trigonometri

Analisis Performa

Kinerja Brent's Method:

1. Kecepatan Konvergensi
   - Konvergensi superlinear
   - Lebih cepat dari bisection
   - Tidak secepat Newton-Raphson (yang butuh turunan)
2. Stabilitas
   - Sangat stabil karena mempertahankan bracket
   - Selalu mengandung akar
3. Robustness
   - Dapat menangani berbagai jenis fungsi
   - Tidak memerlukan informasi turunan

Keunggulan & Keterbatasan

Keunggulan

- Konvergensi dijamin (guaranteed convergence)
- Tidak memerlukan turunan fungsi
- Adaptif dalam memilih strategi optimal
- Kombinasi optimal antara kecepatan dan keandalan

Keterbatasan

- Memerlukan interval awal yang mengandung akar
- Lebih kompleks dibandingkan metode sederhana
- Tidak secepat Newton-Raphson untuk fungsi yang smooth

Kesimpulan

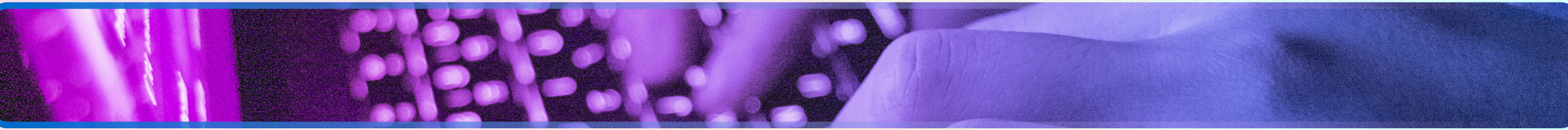
Hasil Implementasi

Hasil Implementasi Brent's Method:

1. Konvergensi cepat dan stabil untuk berbagai jenis fungsi
2. Algoritma adaptif secara otomatis memilih strategi terbaik
3. Sangat cocok untuk aplikasi praktis yang memerlukan keandalan tinggi
4. Implementasi C++ menunjukkan efisiensi komputasi yang baik

Rekomendasi

Brent's Method merupakan pilihan sangat baik untuk aplikasi komputasi numerik yang memerlukan pencarian akar yang andal dan efisien, terutama ketika informasi turunan fungsi tidak tersedia atau sulit diperoleh.



Thank You.
Thank You.
Thank You.

riveliaaisya@gmail.com

github.com/aisyarivel

081297105434