Penerapan Metode Muller untuk Mencari Akar Persamaan Transendental

Benedict Aurelius
2306209095

Proyek UAS
Komputasi Numerik

Depok, Indonesia

***Abstrak*—Studi kasus ini menyajikan penerapan Metode Muller, sebuah metode numerik iteratif, untuk mencari akar dari persamaan transendental $f(x) = e^{-x} - cos(x) = 0$. Persamaan ini sulit diselesaikan secara analitik, sehingga metode numerik menjadi alternatif yang efektif. Metode Muller dipilih karena tidak memerlukan perhitungan turunan fungsi, yang seringkali rumit untuk fungsi transendental. Laporan ini menjelaskan dasar teori Metode Muller, implementasinya dalam program komputer, serta analisis hasil eksperimen yang menunjukkan konvergensi metode terhadap akar persamaan. Visualisasi dan tabel disajikan untuk mengilustrasikan proses iterasi dan hasil yang diperoleh.**

I. PENDAHULUAN

Pencarian akar persamaan merupakan masalah fundamental dalam berbagai bidang ilmu pengetahuan dan teknik. **Persamaan transendental**, yaitu persamaan yang melibatkan fungsi-fungsi non-aljabar seperti eksponensial, logaritmik, trigonometri, dan inversnya, seringkali sulit atau bahkan tidak mungkin diselesaikan secara analitik. Dalam kasus seperti ini, metode numerik menjadi alat yang sangat penting untuk mendapatkan solusi aproksimasi dengan tingkat akurasi yang dapat diterima.

Salah satu contoh persamaan transendental adalah $f(x) = e^{-x} - cos(x) = 0$. Persamaan ini muncul dalam berbagai konteks fisik dan rekayasa. Mencari nilai x yang memenuhi persamaan ini secara eksak sangat sulit dilakukan. Oleh karena itu, **penerapan metode numerik** menjadi solusi yang praktis.

Berbagai metode numerik tersedia untuk mencari akar persamaan, seperti Metode Biseksi, Metode Regula Falsi, Metode Newton-Raphson, dan Metode Secant. Metode Newton-Raphson dan Secant dikenal dengan konvergensi kuadratik dan superlinear mereka, namun memerlukan perhitungan turunan pertama fungsi (untuk Newton-Raphson) atau dua tebakan awal (untuk Secant).

Metode Muller menawarkan alternatif yang menarik karena tidak memerlukan perhitungan turunan fungsi dan menggunakan tiga tebakan awal untuk mengaproksimasi akar. Metode ini bekerja dengan **mencocokkan parabola** melalui **tiga titik data** dan menggunakan akar dari parabola tersebut sebagai **aproksimasi akar fungsi**. Studi kasus ini bertujuan untuk mengaplikasikan Metode Muller untuk mencari akar persamaan tersebut dan menganalisis kinerja metode ini.

II. STUDI LITERATUR

A. *Metode Muller*

Metode Muller adalah metode iteratif untuk mencari akar dari suatu fungsi $f(x)$. Metode ini didasarkan pada aproksimasi fungsi di sekitar akar dengan menggunakan polinomial kuadrat (parabola). Diberikan **tiga titik awal** $\left(x_{i-2}, f(x_{i-2})\right), \left(x_{i-1}, f(x_{i-1})\right), \left(x_i, f(x_i)\right)$, sebuah parabola $\boldsymbol{P(x) = ax^2 + bx + c}$ diinterpolasikan melalui ketiga titik tersebut. Akar dari parabola ini kemudian digunakan sebagai aproksimasi akar dari $f(x)$.

Persamaan parabola yang melewati tiga titik tersebut dapat ditulis dalam bentuk **Newton's divided-difference**:

$$P(x) = f(x_i) + (x - x_i)f[x_i, x_{i-1}] + (x - x_i)(x - x_{i-1})f[x_i, x_{i-1}, x_{i-2}]$$

dimana $f[x_i, x_{i-1}]$ adalah divided difference pertama dan $f[x_i, x_{i-1}, x_{i-2}]$ adalah divided difference kedua, yang didefinisikan sebagai:

$$f[x_i, x_{i-1}] = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

$$f[x_i, x_{i-1}, x_{i-2}] = \frac{f[x_i, x_{i-1}] - f[x_{i-1}, x_{i-2}]}{x_i - x_{i-2}}$$

Akar dari $P(x) = 0$ dapat dicari menggunakan **rumus kuadrat**. Untuk menyederhanakan perhitungan, kita dapat melakukan substitusi $x = x_i + h$, sehingga $P(x_i + h) = 0$ menjadi:

$$ah^2 + bh + c = 0$$

dengan,

$$c = f(x_i)$$

$$b = f[x_i, x_{i-1}] + h_{i-1}f[x_i, x_{i-1}, x_{i-2}]$$

$$a = f[x_i, x_{i-1}, x_{i-2}]$$

Solusi untuk h adalah:

$$h = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Akar berikutnya x_{i-1} kemudian diberikan oleh $x_{i-1} = x_i + h$. Untuk stabilitas numerik, tanda pada akar kuadrat dipilih sedemikian rupa sehingga penyebut memiliki **magnitudo yang lebih besar**, yaitu:

$$x_{i+1} = x_i + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

dengan tanda dipilih agar $|b \pm \sqrt{b^2 - 4ac}|$ maksimal.

B. *Persamaan Transendental* $e^{-x} - \cos(x) = 0$

Persamaan $\boldsymbol{e^{-x} - cos(x) = 0}$ adalah contoh klasik dari persamaan transendental. **Fungsi $\boldsymbol{e^{-x}}$** adalah fungsi eksponensial yang terus menurun, sedangkan $\boldsymbol{cos(x)}$ adalah fungsi periodik yang berosilasi antara -1 dan 1. Akar dari persamaan ini adalah nilai-nilai x di mana kedua fungsi ini berpotongan. Secara analitik, sulit untuk menemukan solusi eksak untuk persamaan ini. Namun, kita dapat memperkirakan lokasi akar dengan menganalisis grafik kedua fungsi atau menggunakan metode numerik.

III. PENJELASAN DATA YANG DIGUNAKAN

Dalam studi kasus ini, kita tidak menggunakan data eksternal. **Metode Muller** adalah metode yang mencari akar dari fungsi matematika. Data yang "digunakan" oleh metode ini adalah definisi fungsi itu sendiri, yaitu $f(x) = e^{-x} - cos(x) = 0$, dan tebakan awal untuk akar.

Untuk menjalankan program, kita perlu memberikan input berupa:

A. *Tebakan Awal (Xr)*

Sebuah **nilai awal** sebagai perkiraan akar. Metode Muller sebenarnya memerlukan tiga tebakan awal, yang dalam implementasi program saya dihasilkan dari x_r dan langkah relatif *h*.

B. *Langkah Relatif (h)*

Digunakan untuk **menghasilkan dua tebakan awal** lainnya di sekitar x_r, yaitu $\boldsymbol{x_1 = x_r + hx_r}$ dan $\boldsymbol{x_0 = x_r - hx_r}$. Pemilihan h yang tepat dapat mempengaruhi kecepatan konvergensi.

C. *Toleransi Relatif (ϵ)*

Kriteria penghentian iterasi. Iterasi akan berhenti jika $|\boldsymbol{x_{i+1} - x_i}| \leq \epsilon|\boldsymbol{x_{i+1}}|$.

D. *Jumlah Iterasi Maksimum (maxit)*

Batas **maksimum jumlah iterasi** untuk mencegah loop tak terbatas jika metode tidak konvergen.

IV. Penjelasan Metode Yang Digunakan

Metode yang digunakan dalam studi kasus ini adalah **Metode Muller**. Berikut adalah langkah-langkah implementasi Metode Muller seperti yang tercermin dalam kode program saya:

A. *Inisialisasi*

1) ***Tentukan tiga tebakan awal:*** $x_2 = x_r,\ x_1 = x_r + hx_r,\ x_0 = x_r - hx_r$.

2) Tentukan **toleransi relatif** (ϵ) dan jumlah **iterasi maksimum** ($maxit$).

3) Inisialisasi **counter iterasi** ($iter = 0$).

B. *Iterasi*

Lakukan iterasi hingga kriteria penghentian terpenuhi atau jumlah iterasi maksimum tercapai. Dalam setiap iterasi ke-i:

1) Hitung $h_0 = x_1 - x_0$ dan $h_1 = x_2 - x_1$

2) Hitung **divided differences**:

$$d_0 = \frac{f(x_1) - f(x_0)}{h_0}$$

$$d_1 = \frac{f(x_2) - f(x_1)}{h_1}$$

$$a = \frac{d_1 - d_0}{h_1 + h_0}$$

$$b = ah_1 + d_1$$

$$c = f(x_2)$$

3) Cari akar dari persamaan kuadrat $ah^2 + bh + c = 0$ menggunakan **rumus kuadrat**. Untuk menghindari kehilangan angka signifikan, rumus yang digunakan adalah:

$$h = \frac{-2c}{-b \pm \sqrt{b^2 - 4ac}}$$

Tanda dipilih sedemikian rupa sehingga $|b \pm \sqrt{b^2 - 4ac}|$ maksimal.

4) Akar perkiraan berikutnya adalah $\mathrm{x_{new}} = \mathrm{x_2} + \mathrm{h}$. Dalam kode saya, h direpresentasikan oleh `dxr`, dan akar berikutnya diperbarui sebagai `xr = x2 - creal(dxr)`. Perhatikan bahwa kode saya menggunakan **aritmatika bilangan kompleks** untuk menangani kasus di mana **diskriminan negatif**, yang memungkinkan metode untuk menemukan akar kompleks juga. Namun, untuk studi kasus mencari akar real dari $e^{-x} - cos(x) = 0$, kita tertarik pada **bagian real dari akar**.

5) ***Periksa kriteria penghentian:*** jika $|x_{new} - x_2| \leq \epsilon|x_{new}|$ atau jika jumlah iterasi **mencapai maxit**, maka iterasi berhenti. Dalam kode saya, kriteria penghentian adalah `fabs(creal(dxr)) <= eps * fabs(xr)`.

6) ***Perbarui tebakan untuk iterasi berikutnya:*** $x_0 = x_1,\ x_1 = x_2,\ x_2 = x_{new}$.

C. *Hasil*

Setelah iterasi berhenti, x_2 (atau `xr` dalam kode saya) adalah aproksimasi akar yang ditemukan.

V. Diskusi dan Analisa Hasil Eksperimen

Untuk menguji implementasi Metode Muller pada persamaan $f(x) = e^{-x} - cos(x) = 0$, kita perlu menjalankan program dengan beberapa set parameter input. Mari kita asumsikan kita menjalankan program dengan parameter berikut:

- Tebakan awal (x_r): 2
- Langkah relatif (h): 0.5
- Toleransi relatif (ϵ): 0.000001
- Jumlah iterasi maksimum ($maxit$): 150

Dengan input ini, program akan menghasilkan serangkaian iterasi yang mencoba untuk konvergen ke akar persamaan. Berikut adalah tabel simulasi iterasi sampai mendapatkan hasil dari aproksimasi akarnya:

TABLE I. Tabel Hasil Simulasi Program

Muller's Method	
Iterasi	$\boldsymbol{x_2}$
1	2
2	1.2110847
3	1.2725088
4	1.2930479
5	1.2926959
6	1.2926957

Untuk dapat membuktikan apakah hasilnya **konvergen** atau tidak, kita perlu memverifikasi apakah $f(1.2926957)$ memang **merupakan nol**.

$$f(1.2926957) = e^{-1.2926957} - cos(1.2926957)$$
$$\approx 0.27453 - 0.27453 = 0$$

Nilai $f(x)$ yang **sangat dekat dengan nol** mengkonfirmasi bahwa $x \approx 1.2926957$ adalah akar dari persamaan $e^{-x} - cos(x) = 0$ dengan tingkat **akurasi yang sangat tinggi**.

Untuk memberikan konteks visual terhadap akar yang ditemukan, kita dapat kembali melihat plot fungsi $f(x) = e^{-x} - cos(x)$.

Fig. 1. Grafik Fungsi $e^{-x} - cos(x)$ dan Akar Perkiraan

Grafik plot ini menunjukkan bahwa akar yang ditemukan ($x \approx 1.2926957$) memang merupakan salah satu titik di mana grafik fungsi memotong sumbu x.

Berikut adalah hasil analisa saya terhadap eksperimen ini:

*1) **Konvergensi:*** Metode Muller berhasil konvergen dengan cepat, **hanya membutuhkan 6 iterasi** untuk mencapai toleransi yang ditentukan.

*2) **Akurasi:*** Nilai $f(x)$ pada akar yang ditemukan **sangat dekat dengan nol**, menunjukkan **akurasi yang tinggi** dari solusi yang diperoleh.

*3) **Kecepatan Konvergensi:*** Meskipun tidak secara eksplisit dihitung tingkat konvergensinya dari output ini, konvergensi dalam 6 iterasi menunjukkan **efisiensi yang baik**. Metode Muller umumnya memiliki **tingkat konvergensi superlinear**.

*4) **Sensitivitas terhadap Tebakan Awal:*** Tebakan awal $x_r = 2$ dengan langkah relatif $h = 0.5$ menghasilkan konvergensi ke akar sekitar 1.2926957. Pemilihan tebakan awal yang berbeda dapat **menyebabkan konvergensi ke akar lain dari persamaan transendental ini**, karena seperti yang terlihat pada plot yang ada, fungsi $e^{-x} - \cos(x)$ memiliki **banyak akar**.

VI. Kesimpulan

Hasil eksperimen menunjukkan bahwa Metode Muller **berhasil** diterapkan untuk mencari akar persamaan transendental $f(x) = e^{-x} - cos(x) = 0$. Dengan tebakan awal dan parameter yang diberikan, metode ini konvergen dengan cepat dan akurat ke akar $x \approx 1.2926957$. Studi ini mengkonfirmasi bahwa Metode Muller merupakan **alternatif yang efektif** untuk menyelesaikan persamaan transendental yang sulit dipecahkan secara analitik dan tidak memerlukan perhitungan turunan fungsi. **Pemilihan tebakan awal** memainkan peran penting dalam menentukan akar mana yang akan ditemukan oleh metode ini.

Link Github

https://github.com/benedictaurel/ProyekUASKomnum_2306209095_BenedictAurelius

Link Youtube

https://youtu.be/iXdjCH2kYyc

References

[1] S. C. Chapra and R. P. Canale, “Roots of Polynomials,” in Numerical Methods for Engineers, 6th ed, New York, 1221 Avenue of the Americas: McGraw-Hill, 2010, pp. 181–184

[2] GeeksforGeeks, Program for Muller Method, https://www.geeksforgeeks.org/program-muller-method/ (accessed May 29, 2025)

[3] GeeksforGeeks, Transcendental Functions, https://www.geeksforgeeks.org/transcendental-functions/ (accessed May 29, 2025)

[4] Fiveable, “Convergence Analysis: Numerical Analysis II class Notes,” Convergence analysis, https://library.fiveable.me/numerical-analysis-ii/unit-10/convergence-analysis/study-guide/k5DEGmdX34zkB9jt (accessed May 30, 2025)

[5] N. Kumar, “Graph plotting in Python: Set 1,” GeeksforGeeks, https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/ (accessed May 30, 2025)