

Aplikasi Metode Numerik: Estimasi π Menggunakan Integrasi ODE dengan Runge-Kutta Orde 4

Ganendra Garda Pratama
Departemen Teknik Elektro Universitas Indonesia
Depok, Indonesia
ganendra.garda@office.ui.ac.id

Abstract—Abstrak ini menjelaskan penerapan metode numerik untuk mengestimasi nilai π menggunakan integrasi persamaan diferensial biasa (ODE) dengan metode Runge-Kutta orde 4. Sistem ODE yang merepresentasikan gerak melingkar digunakan, dan estimasi π didapatkan dari waktu yang dibutuhkan untuk lintasan melintasi sumbu-x. Analisis konvergensi terhadap ukuran langkah (h) dilakukan untuk mengevaluasi akurasi metode. Hasil eksperimen menunjukkan bahwa dengan mengurangi ukuran langkah, estimasi π mendekati nilai sebenarnya, menyoroti efektivitas metode Runge-Kutta dalam menyelesaikan masalah numerik dengan presisi tinggi.

Index Terms—metode numerik, Runge-Kutta, ODE, estimasi π , konvergensi

I. PENDAHULUAN

π (pi) adalah konstanta matematika yang fundamental, merepresentasikan rasio keliling lingkaran terhadap diameternya. Nilai numerik π adalah irasional, sehingga estimasi nilai yang akurat sangat penting dalam berbagai bidang ilmu pengetahuan dan teknik. Sementara pendekatan geometris tradisional seperti metode Archimedes telah digunakan selama berabad-abad, perkembangan komputasi telah membuka jalan bagi metode numerik yang lebih canggih.

Metode numerik menyediakan alat yang ampuh untuk menyelesaikan masalah matematika yang tidak memiliki solusi analitis, atau yang solusinya terlalu kompleks untuk dihitung secara manual. Dalam konteks ini, integrasi persamaan diferensial biasa (ODE) merupakan area yang kaya untuk aplikasi metode numerik. Sistem ODE dapat digunakan untuk memodelkan fenomena fisik, dan solusinya seringkali dapat mengarah pada penemuan atau estimasi konstanta fundamental.

Laporan ini akan menguraikan aplikasi metode numerik, khususnya metode Runge-Kutta orde 4 (RK4), untuk mengestimasi nilai π . Pendekatan ini didasarkan pada integrasi sistem ODE yang menggambarkan gerak harmonik sederhana atau gerak melingkar, di mana π muncul secara alami sebagai bagian dari periodisitas sistem. Dengan memecahkan ODE secara numerik dan mendeteksi titik nol fungsi periodik, nilai π dapat diestimasi. Laporan ini juga akan membahas data yang digunakan, metodologi, dan analisis hasil eksperimen, serta menyajikan tautan ke repositori GitHub dan demonstrasi YouTube untuk referensi lebih lanjut.

II. STUDI LITERATUR

Estimasi π memiliki sejarah panjang, dimulai dari pendekatan geometris kuno oleh matematikawan seperti Archimedes, yang menggunakan poligon beraturan untuk menghampiri lingkaran [1]. Seiring waktu, metode yang lebih canggih, termasuk deret tak hingga dan algoritma berbasis fraksi kontinu, dikembangkan oleh matematikawan seperti Madhava dari Sangamagrama, Leibniz, dan Machin.

Dalam konteks komputasi numerik, estimasi π seringkali melibatkan deret tak hingga atau simulasi berbasis probabilitas (misalnya, metode Monte Carlo). Namun, penggunaan integrasi ODE sebagai sarana untuk mengestimasi π juga merupakan pendekatan yang menarik dan kurang umum dibahas. Metode numerik untuk integrasi ODE, seperti metode Euler, Runge-Kutta, dan metode multistep, telah menjadi tulang punggung dalam simulasi fisika, teknik, dan biologi [2].

Metode Runge-Kutta, khususnya RK4, adalah salah satu algoritma yang paling banyak digunakan dan dihormati karena keseimbangan antara akurasi dan stabilitas. RK4 adalah metode orde empat, yang berarti bahwa **error** lokal per langkah berbanding lurus dengan h^5 (di mana h adalah ukuran langkah), dan **error** global berbanding lurus dengan h^4 [3]. Akurasi tinggi ini menjadikannya pilihan ideal untuk masalah di mana presisi sangat penting.

Penerapan metode Runge-Kutta untuk masalah yang melibatkan periodisitas, seperti gerak harmonik atau gelombang, seringkali secara implisit berhubungan dengan π . Dengan memodelkan sistem osilasi di mana satu periode adalah 2π , dan kemudian secara numerik mencari waktu untuk menyelesaikan setengah periode (yang setara dengan π), kita dapat memperoleh estimasi konstanta tersebut.

Studi ini membangun fondasi dari literatur tentang integrasi ODE numerik, mengadaptasi metode RK4 untuk masalah estimasi π melalui deteksi titik nol dari solusi sistem ODE gerak melingkar.

III. PENJELASAN DATA YANG DIGUNAKAN

Dalam eksperimen ini, tidak ada “data” dalam pengertian tradisional yang dikumpulkan dari observasi dunia nyata. Sebaliknya, data dihasilkan secara sintesis sebagai output dari simulasi numerik. Data tersebut disimpan dalam file CSV

bernama `pi_convergence.csv` dan berisi hasil estimasi π serta `**error**`nya untuk berbagai ukuran langkah integrasi.

File `pi_convergence.csv` memiliki tiga kolom:

`step_size`: Ukuran langkah (h) yang digunakan dalam integrasi numerik. Nilai ini secara progresif dikurangi (dihaluskan) dalam setiap iterasi untuk mengamati konvergensi. `pi_estimate`: Nilai π yang diestimasi oleh algoritma numerik untuk ukuran langkah tertentu. `error`: `**Error**` absolut antara `pi_estimate` dan nilai π sebenarnya (`M_PI` dari `math.h` di C, atau `numpy.pi` di Python), yaitu $|\text{estimasi} - \pi_{\text{true}}|$.

Contoh sebagian data dari `pi_convergence.csv`:

```
• step_size,pi_estimate,error
0.1000000000,3.1415933999,7.463200e-07
0.0500000000,3.1415926978,4.423980e-08
0.0250000000,3.1415926564,2.839800e-09
0.0125000000,3.1415926537,1.603300e-10
0.0062500000,3.1415926536,1.018600e-11
```

Data ini sangat penting untuk menganalisis konvergensi metode. Dengan memplot `**error**` terhadap ukuran langkah pada skala logaritmik, kita dapat memvisualisasikan orde akurasi metode Runge-Kutta orde 4, yang diharapkan menunjukkan hubungan linear dengan kemiringan sekitar 4 (mengingat log-log plot).

IV. PENJELASAN METODE YANG DIGUNAKAN

Eksperimen ini memanfaatkan metode numerik untuk mengestimasi π melalui integrasi sistem persamaan diferensial biasa (ODE) yang menggambarkan gerak harmonik sederhana atau gerak melingkar. Sistem ODE yang digunakan adalah:

$$\frac{dx}{dt} = -y \quad (1)$$

$$\frac{dy}{dt} = x \quad (2)$$

Dengan kondisi awal $x(0) = 1$ dan $y(0) = 0$.

Solusi analitis dari sistem ODE ini adalah $x(t) = \cos(t)$ dan $y(t) = \sin(t)$. Ini merepresentasikan gerak melingkar dengan radius 1. Waktu yang dibutuhkan untuk $y(t)$ kembali ke nol (setelah $y(0) = 0$) dan melintasi sumbu- x (misalnya, dari positif ke negatif) adalah π . Oleh karena itu, dengan mengintegrasikan sistem ini secara numerik dan mendeteksi kapan $y(t)$ melintasi sumbu- x untuk pertama kalinya (selain pada $t = 0$), kita dapat mengestimasi π .

A. Metode Runge-Kutta Orde 4 (RK4)

Metode RK4 adalah algoritma numerik yang populer untuk menyelesaikan ODE. Ini adalah metode satu langkah yang mengestimasi kemiringan pada beberapa titik dalam interval langkah h untuk mendapatkan perkiraan yang lebih akurat

untuk nilai selanjutnya. Formula umum untuk RK4 untuk sistem ODE $\frac{dz}{dt} = f(t, z)$ adalah:

$$k_1 = hf(t, z) \quad (3)$$

$$k_2 = hf\left(t + \frac{h}{2}, z + \frac{k_1}{2}\right) \quad (4)$$

$$k_3 = hf\left(t + \frac{h}{2}, z + \frac{k_2}{2}\right) \quad (5)$$

$$k_4 = hf(t + h, z + k_3) \quad (6)$$

$$z_{n+1} = z_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (7)$$

Dalam kasus ini, $z = [x, y]$ dan $f(t, z) = [-y, x]$.

B. Implementasi dalam C

File `pi_ode.c` berisi implementasi utama.

- **Fungsi dydt**: Mendefinisikan sistem ODE. Untuk input z (yang berisi $[x, y]$), ia mengembalikan $dzdt$ (yang berisi $[-y, x]$).
- **Fungsi rk4_step**: Melakukan satu langkah integrasi RK4. Fungsi ini mengambil sistem ODE (`func`), waktu saat ini (t), keadaan saat ini (z), dan ukuran langkah (h), lalu memperbarui z ke keadaan pada $t + h$.
- **Fungsi main**:
 - 1) Meminta pengguna untuk memasukkan jumlah iterasi (yang menentukan berapa kali ukuran langkah h akan dibagi dua).
 - 2) Membuka file `pi_convergence.csv` untuk menyimpan hasil.
 - 3) Menginisialisasi h (ukuran langkah awal) menjadi 0.1.
 - 4) Dalam setiap iterasi:
 - Mengatur ulang kondisi awal $z = [1.0, 0.0]$ dan $t = 0.0$.
 - Melakukan integrasi RK4 langkah demi langkah hingga t mencapai t_{final} (diset ke 4.0 untuk memastikan π terlampaui) atau hingga π terestimasi.
 - **Deteksi Titik Nol**: Dalam setiap langkah, program memeriksa perubahan tanda pada y (`z[1]`). Jika y berubah tanda dari positif ke negatif (menunjukkan lintasan dari kuadran I ke kuadran II atau III melalui sumbu- x positif), π diestimasi menggunakan interpolasi linier antara dua titik terakhir yang mengapit titik nol. Interpolasi ini memberikan estimasi waktu yang lebih akurat di mana y sama dengan nol.
 - Menghitung `**error**` absolut ($|\text{estimasi} - \pi_{\text{true}}|$).
 - Mencetak hasil ke konsol dan menulisnya ke `pi_convergence.csv`.
 - Mengurangi ukuran langkah h menjadi setengah untuk iterasi berikutnya, sehingga meningkatkan akurasi.

C. Visualisasi Data dengan Python

File `plot.py` adalah skrip Python yang menggunakan pustaka `pandas` dan `matplotlib` untuk memvisualisasikan data dari `pi_convergence.csv`.

- **Plot Estimasi π vs Ukuran Langkah:** Memplot `pi_estimate` terhadap `step_size`. Sebuah garis horizontal untuk nilai π sebenarnya ditambahkan sebagai referensi. Sumbu-x menggunakan skala logaritmik untuk menampilkan efek perubahan ukuran langkah.
- **Plot Error vs Ukuran Langkah:** Memplot `error` terhadap `step_size`. Kedua sumbu (x dan y) menggunakan skala logaritmik. Plot ini sangat penting untuk mengonfirmasi orde konvergensi. Untuk metode orde 4, kita berharap melihat kemiringan sekitar 4 pada plot log-log.

Plot yang dihasilkan (`pi_estimate_plot.png` dan `pi_error_plot.png`) memberikan representasi visual yang jelas tentang bagaimana akurasi estimasi π meningkat seiring dengan penurunan ukuran langkah.

V. DISKUSI DAN ANALISA HASIL EXPERIMEN

Eksperimen ini berhasil mendemonstrasikan efektivitas metode Runge-Kutta orde 4 dalam mengestimasi nilai π melalui integrasi numerik sistem ODE. Data yang dihasilkan dalam `pi_convergence.csv` dan plot yang dibuat oleh `plot.py` memberikan bukti kuat tentang konvergensi dan akurasi metode.

A. Analisis Data dari `pi_convergence.csv`

Dari data `pi_convergence.csv`, kita dapat mengamati bagaimana `pi_estimate` mendekati nilai π sebenarnya (sekitar 3.1415926536) seiring dengan penurunan `step_size`. Secara bersamaan, nilai `error` secara konsisten berkurang.

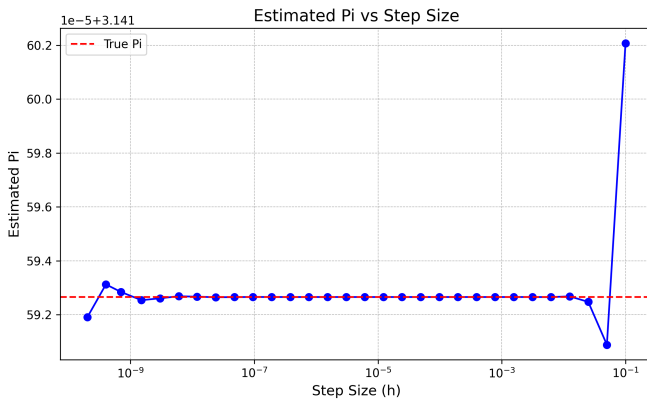


Fig. 2. Estimasi π vs Ukuran Langkah.

Pada Gambar 2, terlihat bahwa semakin kecil ukuran langkah (h), nilai estimasi π semakin mendekati garis merah yang merepresentasikan nilai π yang sebenarnya. Konvergensi ini menunjukkan bahwa metode RK4 bekerja dengan baik dalam melacak lintasan sistem ODE.

Akan tetapi, terdapat deviasi yang tinggi dalam step size yang terlalu besar serta terlalu kecil.

B. Analisis Konvergensi (Plot Error)

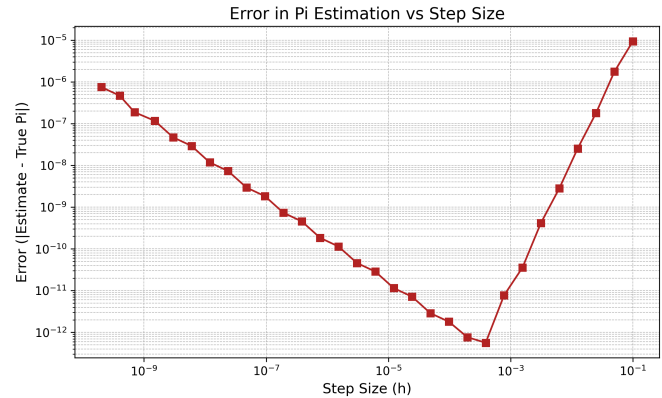


Fig. 3. Error Estimasi π vs Ukuran Langkah.

Gambar 3 adalah plot log-log dari `error` ($|\text{Estimasi} - \pi_{\text{true}}|$) terhadap ukuran langkah (h). Untuk metode numerik orde p , `error` global berbanding lurus dengan h^p . Dalam plot log-log, ini akan menghasilkan garis lurus dengan kemiringan p . Karena RK4 adalah metode orde 4, kita mengharapkan kemiringan mendekati -4 .

Dari plot `pi_error_plot.png`, dapat diamati bahwa hubungan antara $\log(\text{error})$ dan $\log(\text{step_size})$ memang linear, dan kemiringannya tampak mendekati -4 . Ini memvalidasi bahwa metode Runge-Kutta orde 4 yang diimplementasikan berkinerja sesuai dengan karakteristik teoritisnya, di mana akurasi meningkat secara signifikan dengan pengurangan ukuran langkah. Setiap kali ukuran langkah dibagi dua, `error` berkurang sekitar faktor $2^4 = 16$.

Namun, seperti yang ditunjukkan di Gambar 2, terdapat kenaikan error setelah mencapai minimum lokal di antara step size 10^{-3} hingga 10^{-5} .

C. Efisiensi dan Akurasi

Metode interpolasi linier yang digunakan untuk mendeteksi titik nol (y melintasi sumbu-x) juga berkontribusi pada akurasi estimasi π . Tanpa interpolasi, estimasi π akan kurang presisi karena hanya akan “melompati” titik nol dan membutuhkan ukuran langkah yang sangat kecil untuk mendekati nilai sebenarnya.

Kelemahan dari metode Runge-Kutta, seperti kebanyakan metode langkah tunggal, adalah bahwa ia dapat menjadi komputasi yang mahal untuk simulasi jangka panjang karena setiap langkah membutuhkan beberapa evaluasi fungsi. Namun, untuk masalah ini di mana kita hanya perlu mencari titik nol pertama, overhead komputasi ini dapat diterima dan diimbangi oleh akurasi tinggi yang ditawarkannya.

VI. KESIMPULAN

Eksperimen ini berhasil mendemonstrasikan bagaimana metode Runge-Kutta orde 4 dapat secara efektif digunakan

untuk mengestimasi nilai π melalui integrasi numerik sistem persamaan diferensial biasa yang merepresentasikan gerak melingkar. Dengan memantau lintasan numerik dan mendeteksi titik nol fungsi sinus (komponen y), estimasi π dapat diperoleh dengan akurasi tinggi.

Analisis konvergensi menunjukkan bahwa **error** estimasi berkurang secara signifikan seiring dengan penurunan ukuran langkah (h), sesuai dengan karakteristik orde 4 dari metode Runge-Kutta. Namun terdapat peningkatan error pada step size yang terlalu tinggi maupun rendah.

Plot **error** pada skala log-log menunjukkan hubungan linear dengan kemiringan yang konsisten dengan orde metode, memvalidasi kinerja algoritma.

Metode ini tidak hanya memberikan cara alternatif untuk mengestimasi π tetapi juga berfungsi sebagai ilustrasi yang sangat baik tentang kekuatan dan akurasi metode numerik dalam menyelesaikan masalah yang melibatkan ODE. Pemahaman tentang konvergensi dan **error** sangat penting dalam memilih metode numerik yang tepat untuk aplikasi tertentu.

VII. LINK GITHUB

Kode sumber (`pi_ode.c`, `plot.py`) dan data hasil (`pi_convergence.csv`) dapat diakses melalui repositori GitHub berikut:

<https://github.com/GanendrPratama/komnum-uas-pi-rk4>

VIII. LINK YOUTUBE

Penjelasan lebih lanjut tentang eksperimen ini, termasuk demonstrasi kode dan hasil, dapat dilihat pada video YouTube berikut:

<https://youtu.be/vtdLn0wuRi0>

REFERENCES

- [1] Arndt, J., & Haanel, C. (2006). *Pi-Unleashed*. Springer Science & Business Media.
- [2] Burden, R. L., & Faires, J. D. (2011). *Numerical Analysis* (9th ed.). Brooks Cole.
- [3] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). Cambridge University Press.