Laporan Tugas Metode Numerik:

Integrasi Romberg

Oleh:

[Nama Anda]
NIM: [NIM Anda]

Program Studi [Program Studi Anda]
Fakultas [Fakultas Anda]
Universitas [Universitas Anda]

9 November 2025

Daftar Isi

1 Pendahuluan

Integrasi Romberg adalah sebuah metode numerik yang digunakan untuk mengestimasi nilai integral tentu. Metode ini pada dasarnya adalah penerapan dari Ekstrapolasi Richardson pada hasil-hasil yang diperoleh dari Aturan Trapesium Komposit.

Aturan Trapesium memiliki galat (error) $O(h^2)$. Dengan menerapkan ekstrapolasi, kita dapat menggabungkan dua hasil Trapesium (misalnya dengan h dan $h/2$) untuk menghasilkan estimasi baru yang memiliki orde galat lebih baik, yaitu $O(h^4)$. Jika proses ini dilanjutkan, kita bisa mendapatkan orde galat $O(h^6)$, $O(h^8)$, dan seterusnya. Proses ini membangun sebuah tabel segitiga, di mana setiap kolom baru memiliki akurasi yang lebih tinggi daripada sebelumnya.

2 Metodologi

Dalam laporan ini, dua pendekatan komputasi digunakan untuk menyelesaikan soal integral yang diberikan:

2.1 Python Step-by-Step (romberg_step_by_step.py)

Sebuah skrip Python dibuat dari nol untuk mengimplementasikan algoritma Romberg. Pendekatan ini dimulai dengan fungsi aturan Trapesium komposit. Kemudian, skrip ini secara eksplisit menghitung setiap entri dalam tabel Romberg ($R_{0,0}$, $R_{1,0}$, $R_{1,1}$, dst.) langkah demi langkah. Tujuannya adalah untuk menunjukkan pemahaman proses dan transparansi algoritma.

2.2 Scipy Library (romberg_efisien.py)

Pendekatan ini menggunakan fungsi `scipy.integrate.romberg` yang sudah teroptimasi. Fungsi ini adalah "black box" yang efisien, secara otomatis melakukan ekstrapolasi hingga mencapai toleransi galat yang diinginkan. Ini digunakan untuk memvalidasi hasil dan menunjukkan metode komputasi yang efisien di dunia nyata.

3 Pembahasan Soal 1: $I = \int_0^{\pi/2} \cos(x)\, dx$

3.1 Hasil Perhitungan Step-by-Step

Berikut adalah output konsol dari skrip `romberg_step_by_step.py` yang menunjukkan pembangunan tabel hingga $R_{2,2}$:

```
===== SOAL 1: integral(cos(x), 0, pi/2) =====

--------------------------------------------------
Memulai Perhitungan Romberg untuk level R(2,2)
--------------------------------------------------
--- Kolom Pertama (m=0): Aturan Trapesium ---
R(0, 0) [n= 1]: 0.7853981634
R(1, 0) [n= 2]: 0.9480594490
R(2, 0) [n= 4]: 0.9871158010
--- Kolom Ekstrapolasi (m=1) [O(h^4)] ---
R(1, 1): 1.0022795408
R(2, 1): 1.0001345849
--- Kolom Ekstrapolasi (m=2) [O(h^6)] ---
R(2, 2): 0.9999913657
--- Hasil Tabel Segitiga Romberg (R[k,m]) ---
0.7853981634
0.9480594490 1.0022795408
0.9871158010 1.0001345849 0.9999913657
>>> Hasil Akhir R(2,2): 0.9999913657
==================================================
```

3.2 Perbandingan dengan Nilai Eksak

Nilai eksak dari integral ini dapat dihitung secara analitis:

$$I = \int_0^{\pi/2} \cos(x)\,dx = [\sin(x)]_0^{\pi/2} = \sin\left(\frac{\pi}{2}\right) - \sin(0) = 1 - 0 = 1 \tag{1}$$

- Hasil $R_{0,0}$ (Trapesium, $n = 1$): 0.785398 (Galat $\approx 21.5\%$)
- Hasil $R_{2,0}$ (Trapesium, $n = 4$): 0.987116 (Galat $\approx 1.29\%$)
- Hasil $R_{2,2}$ (Romberg $O(h^6)$): 0.9999913657

Hasil $R_{2,2}$ memiliki galat absolut sebesar $|1 - 0.9999913657| \approx 8.63 \times 10^{-6}$, yang menunjukkan peningkatan akurasi yang sangat signifikan.

4 Pembahasan Soal 2: $I = \int_0^1 x^2\,dx$

4.1 Hasil Perhitungan Step-by-Step

Berikut adalah output konsol dari skrip `romberg_step_by_step.py` untuk soal kedua:

```
===== SOAL 2: integral(x^2, 0, 1) =====
--------------------------------------------------
Memulai Perhitungan Romberg untuk level R(2,2)
--------------------------------------------------
--- Kolom Pertama (m=0): Aturan Trapesium ---
R(0, 0) [n= 1]: 0.5000000000
R(1, 0) [n= 2]: 0.3750000000
R(2, 0) [n= 4]: 0.3437500000
--- Kolom Ekstrapolasi (m=1) [O(h^4)] ---
R(1, 1): 0.3333333333
R(2, 1): 0.3333333333
--- Kolom Ekstrapolasi (m=2) [O(h^6)] ---
R(2, 2): 0.3333333333
--- Hasil Tabel Segitiga Romberg (R[k,m]) ---
0.5000000000
0.3750000000 0.3333333333
0.3437500000 0.3333333333 0.3333333333
>>> Hasil Akhir R(2,2): 0.3333333333
==================================================
```

4.2 Perbandingan dengan Nilai Eksak

Nilai eksak dari integral ini adalah:

$$I = \int_0^1 x^2\,dx = \left[\frac{x^3}{3}\right]_0^1 = \frac{1^3}{3} - \frac{0^3}{3} = \frac{1}{3} \approx 0.333333\ldots \tag{2}$$

Dalam kasus ini, hasil $R_{1,1}$ (orde $O(h^4)$) sudah memberikan jawaban yang eksak (0.333...). Ini adalah properti yang menarik dari metode ini. Ekstrapolasi pertama (kolom $m = 1$) setara dengan Aturan Simpson 1/3, yang memang eksak untuk polinomial hingga derajat 3. Karena $f(x) = x^2$ adalah polinomial derajat 2, kolom kedua ($R_{k,1}$) dan semua kolom setelahnya akan menghasilkan nilai eksak.

5 Analisis Perbandingan Tool (Transparansi vs. Efisiensi)

Tugas ini menyoroti *trade-off* penting dalam komputasi numerik: transparansi proses vs. efisiensi komputasi.

5.1 Excel

Sangat baik untuk transparansi visual. Kita dapat membangun tabel Romberg sel demi sel dan melihat bagaimana rumus ekstrapolasi menghubungkan setiap nilai. Ini sangat bagus untuk pelaporan visual. Namun, metode ini menjadi tidak praktis dan rentan error jika kita membutuhkan level yang lebih tinggi (misal, $R_{5,5}$ atau $R_{10,10}$), karena memerlukan pengaturan manual yang signifikan.

5.2 scipy.integrate.romberg

Mewakili efisiensi komputasi. Ini adalah tool "black box" yang sangat cepat dan akurat, dioptimalkan dalam bahasa C di baliknya. Dalam praktik profesional, ini adalah pilihan yang jelas. Namun, untuk laporan akademik yang tujuannya adalah menunjukkan pemahaman algoritma, library ini menyembunyikan langkah-langkah internalnya (meskipun `show=True` dapat memberi kita intipan).

5.3 Python Step-by-Step (romberg_step_by_step.py)

Pendekatan ini adalah kompromi terbaik untuk tujuan tugas ini. Pendekatan ini memanfaatkan kekuatan otomatisasi dan presisi Python (mengatasi kelemahan Excel) sekaligus menjaga transparansi algoritma (mengatasi kelemahan "black box" scipy). Dengan mencetak setiap langkah $R(k, m)$, skrip ini membuktikan bahwa kita tidak hanya mendapatkan jawaban yang benar, tetapi kita juga memahami bagaimana jawaban itu didapat melalui proses ekstrapolasi rekursif.

6 Kesimpulan

Metode Integrasi Romberg telah berhasil diimplementasikan dan divalidasi untuk dua kasus integral:

1. $\int_0^{\pi/2} \cos(x)\,dx$ dengan hasil $R_{2,2} = 0.9999913657$ (galat $\approx 8.63 \times 10^{-6}$)
2. $\int_0^1 x^2\,dx$ dengan hasil eksak $R_{1,1} = 0.3333333333$

Metode Romberg menunjukkan konvergensi yang sangat cepat dan efisien, terutama untuk fungsi yang halus dan kontinu. Implementasi step-by-step dalam Python memberikan keseimbangan optimal antara transparansi algoritma dan efisiensi komputasi.