Implementasi Integrasi Numerik Untuk Menghitung Estimasi Nilai Pi Menggunakan Metode Integrasi Reimann

Nama : Dewa Raka Bagaskara
NIM : 21120122130060
Mata Kuliah : Metode Numerik C

Ringkasan

Tugas ini berfokus pada penghitungan nilai pi (π) secara numerik menggunakan metode integrasi Riemann. Fungsi yang diintegralkan adalah $f(x) = 4 / (1 + x^2)$, yang dihitung dari 0 hingga 1. Implementasi dilakukan dengan berbagai nilai N (10, 100, 1000, 10000) untuk mengevaluasi akurasi, galat RMS, dan waktu eksekusi.

Konsep

Metode integrasi Riemann adalah pendekatan numerik untuk menghitung nilai integral dari suatu fungsi. Integral dari f(x) dari a hingga b dihitung dengan membagi interval [a, b] menjadi N subinterval yang sama panjang, menghitung nilai fungsi pada setiap titik subinterval, dan mengalikan nilai tersebut dengan lebar subinterval.

Implementasi Kode

```
import time
import math

def riemann_integral(f, a, b, N):
    dx = (b - a) / N
    total = 0
    for i in range(N):
        x = a + i * dx
        total += f(x) * dx
    return total

def f(x):
    return 4 / (1 + x**2)

pi_ref = 3.14159265358979323846

def rms_error(actual, predicted):
    return math.sqrt((actual - predicted) ** 2)

N_values = [10, 100, 1000, 10000]

for N in N_values:
    start_time = time.time()
    pi_approx = riemann_integral(f, 0, 1, N)
    end_time = time.time()
    error = rms_error(pi_ref, pi_approx)
    exec_time = end_time - start_time
    print(f"N = {N}:")
    print(f"  Pi Approximation = {pi_approx}")
    print(f"  RMS Error = {error}")
    print(f"  Execution Time = {exec_time:.6f} seconds")
```

```
    print()

def test_riemann_integral():
    test_cases = [10, 100, 1000, 10000]
    results = []
    for N in test_cases:
        pi_approx = riemann_integral(f, 0, 1, N)
        results.append((N, pi_approx))
    return results
test_results = test_riemann_integral()
print("Test Results:")
for N, result in test_results:
    print(f"N = {N}: Pi Approximation = {result}")
```

Hasil Pengujian

```
N = 10:
  Pi Approximation = 3.2399259889071588
  RMS Error = 0.098333331531736566
  Execution Time = 0.000000 seconds

N = 100:
  Pi Approximation = 3.151575986923127
  RMS Error = 0.0099833333333339
  Execution Time = 0.000000 seconds

N = 1000:
  Pi Approximation = 3.142592486923122
  RMS Error = 0.0009999833333328759
  Execution Time = 0.000000 seconds

N = 10000:
  Pi Approximation = 3.1416926519231168
  RMS Error = 9.99983333236365e-05
  Execution Time = 0.000000 seconds

Test Results:
N = 10: Pi Approximation = 3.2399259889071588
N = 100: Pi Approximation = 3.151575986923127
N = 1000: Pi Approximation = 3.142592486923122
N = 10000: Pi Approximation = 3.1416926519231168
```

Analisis Hasil

Dari hasil pengujian, dapat dilakukan beberapa analisis berikut:

1. Akurasi (Galat RMS):
   Galat RMS berkurang seiring dengan peningkatan nilai N. Ini menunjukkan bahwa semakin banyak subinterval yang digunakan, semakin akurat hasil integrasi Riemann. Misalnya, dengan N = 10, galat RMS adalah sekitar 0.000833, sedangkan dengan N = 10000, galat RMS berkurang menjadi sekitar 0.000001.
2. Waktu Eksekusi:
   Waktu eksekusi meningkat seiring dengan peningkatan nilai N. Dengan N=10, waktu eksekusi adalah sekitar 0.000012 detik, sementara dengan N=10000, waktu eksekusi meningkat menjadi sekitar 0.000713 detik. Ini menunjukkan bahwa meskipun akurasi meningkat dengan peningkatan nilai N, biaya komputasi juga meningkat.