

test

April 10, 2025

```
[3]: import numpy as np

#
def integrand(x):
    return x * np.log(1 / x)

#
def monte_carlo_integration(num_samples):
    x_samples = np.random.uniform(0, 1, num_samples)
    y_samples = integrand(x_samples)
    integral_estimate = np.mean(y_samples)
    return integral_estimate

#
def dynamic_monte_carlo_integration(initial_samples, tolerance):
    num_samples = initial_samples
    prev_result = monte_carlo_integration(num_samples)
    while True:
        num_samples *= 2 #
        current_result = monte_carlo_integration(num_samples)
        if abs(current_result - prev_result) < tolerance:
            break
        prev_result = current_result
    return current_result, num_samples

#
initial_samples = 1000
tolerance = 1e-6

#
result, final_samples = dynamic_monte_carlo_integration(initial_samples,
    ↪tolerance)
print(f"      : {result:.6f}")
print(f"      : {final_samples}")
```

```
: 0.250006
: 262144000
```