Assignment – Week 3

Azhar Harisandi

22320011

I. Problem

1. Rancanglah algoritma sederhana untuk mencari solusi integral dengan metoda Monte Carlo! Kemudian, tentukanlah nilai integral di bawah ini! (Bandingkan dengan solusi analitik)

   a. $f(x) = \int_1^5 x^2 dx$

   b. $f(x) = \int_1^5 x^2 + 4x \sin x \ dx$

   c. $f(x) = \int_0^4 \sqrt[4]{15x^3 + 21x^2 + 41x + 3}\, e^{-0.5x} dx$

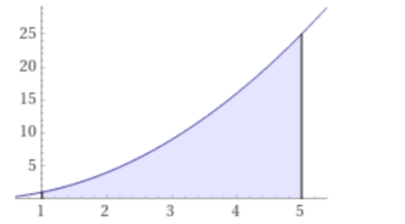
II. Solutions

Analytical Solution

Definite integral:

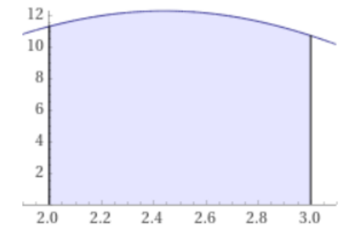
$$\int_1^5 x^2\, dx = \frac{124}{3} \approx 41.333$$

Visual representation of the integral:



$$\int_2^3 \left(x^2 + 4\, x \sin(x)\right) dx = \frac{19}{3} - 4 \sin(2) + 4 \sin(3) + 8 \cos(2) - 12 \cos(3) \approx 11.8114$$

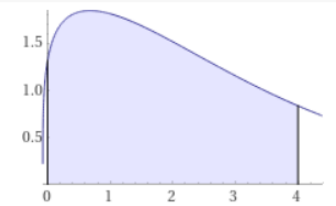
Visual representation of the integral:



Definite integral:

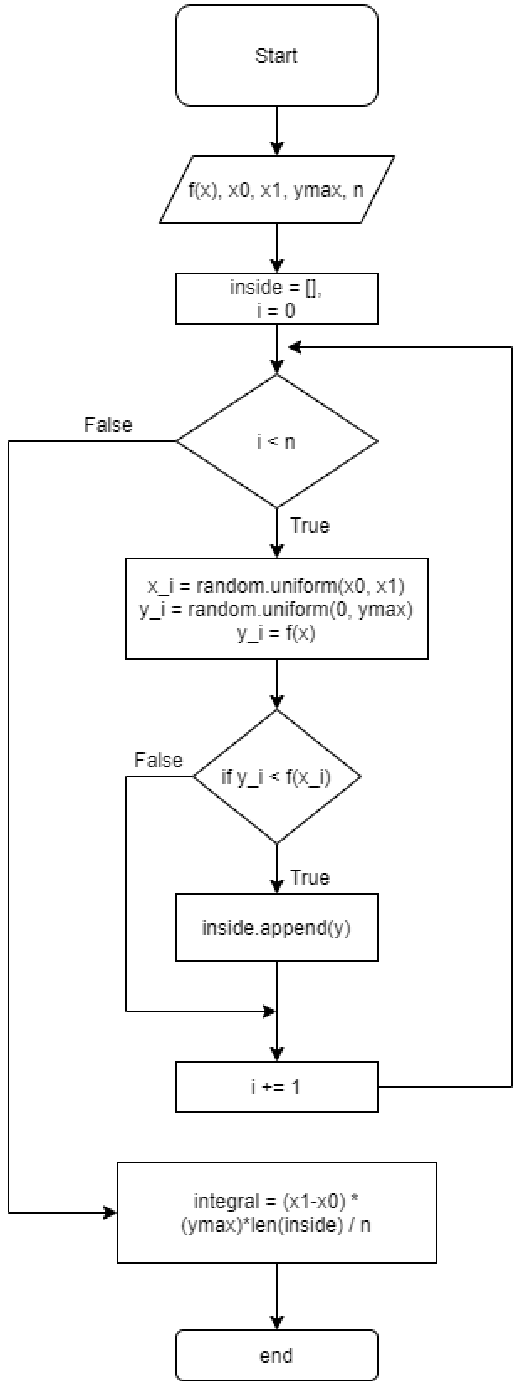
$$\int_0^4 e^{-x/2} \sqrt[4]{3 + 41\, x + 21\, x^2 + 15\, x^3}\; dx \approx 5.76743349069593\ldots$$

Visual representation of the integral:



1. Direct sampling method

1.1 Flowchart for direct sampling method

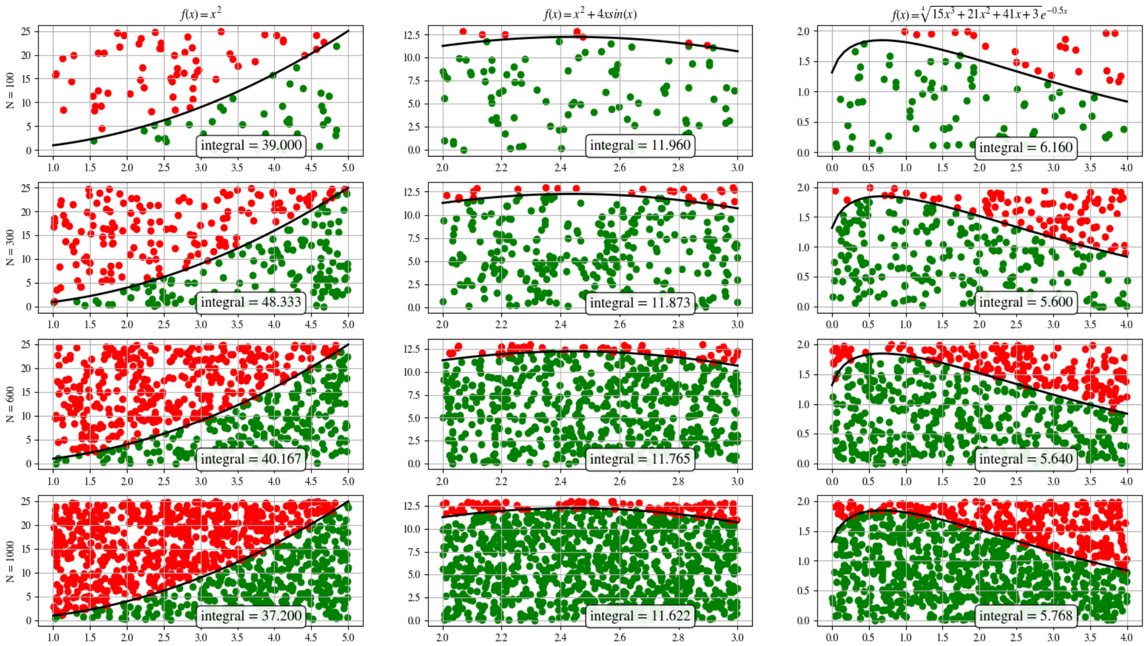


1.2 Code snippets

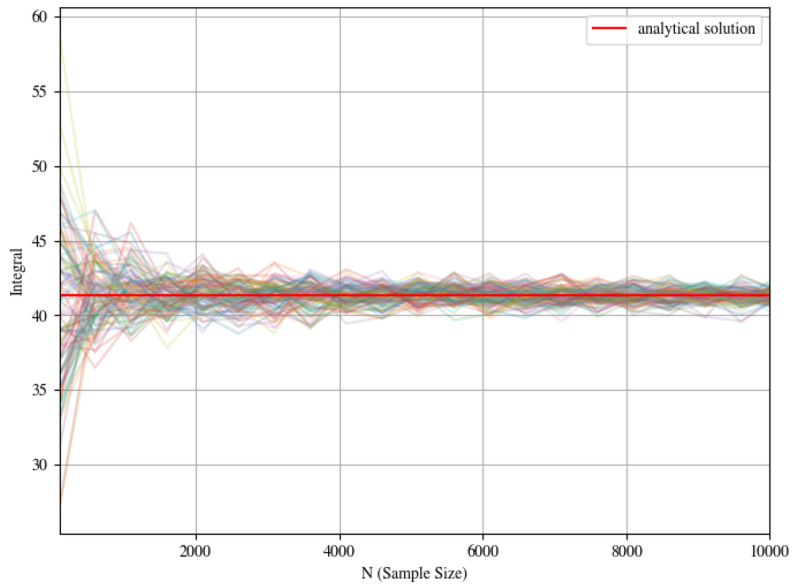
```
def mc_direct_sampling(f, fmax, a, b, N):
    '''
    perform direct sampling using uniform RNG
    f : function of one variable
    fmax : maximum value for f in a given interval
    a, b : lower and upper bound of integration
    N : Sample size
    '''
    hit = {'x': [], 'y': []}
    miss = {'x': [], 'y': []}
    for _ in range(N):
        x = random.uniform(a, b)
        y = random.uniform(0, fmax)
        if y < f(x):
            hit['x'].append(x)
            hit['y'].append(y)
        else:
            miss['x'].append(x)
            miss['y'].append(y)

    return hit, miss, len(hit['x'])/N * (b-a) * fmax
```

1.3 Analisis

Ilustrasi titik hit (hijau) and miss (merah) dari sampel acak :



Karena metode ini memanfaatkan RNG (*Random Number Generator*), maka setiap percobaan/realisasi akan menghasilkan nilai yang berbeda. Berikut adalah contoh ilustrasi dari 100 percobaan/realisasi berbeda untuk nilai sampel dari 100 hingga 10000 untuk fungsi pertama. Dapat dilihat bahwa semakin besar sampel, semakin kecil nilai variansi, dan nilai rata-rata integral dari semua realisasi/percobaan semakin mendekati solusi analitik.

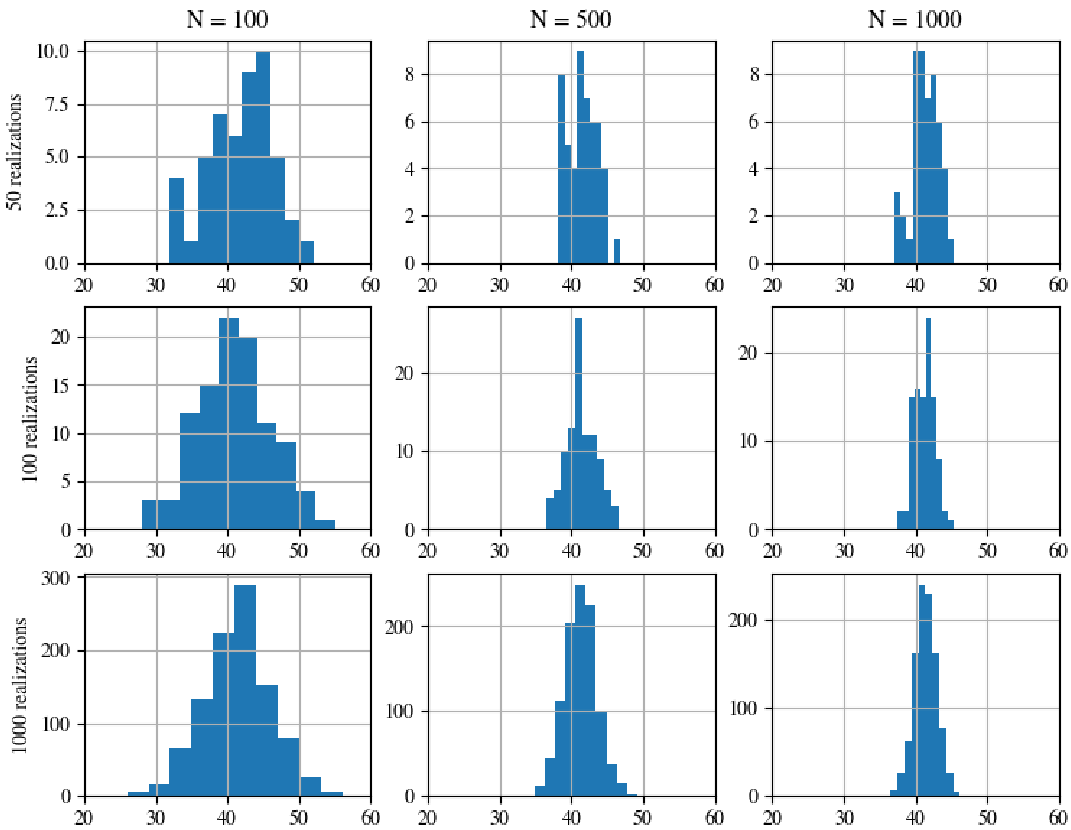


Dalam teori probabilitas hal ini dikenal sebagai ***Law of large number***, yang menyatakan bahwa semakin besar suatu sampel percobaan, nilai rata-rata dari percobaan-percobaan tersebut akan mendekati nilai ekspektasi secara matematis, dalam kasus ini ekspektasi secara matematisnya adalah ekspektasi nilai proporsi dari titik yang ada di bawah kurva terhadap keseluruhan titik dalam satu area.

Hal ini penting untuk metode yang memanfaatkan nilai acak untuk menjamin bahwa nilai rata-rata empiris akan konvergen ke nilai ekspektasi matematisnya seiring dengan bertambahnya jumlah sampel.

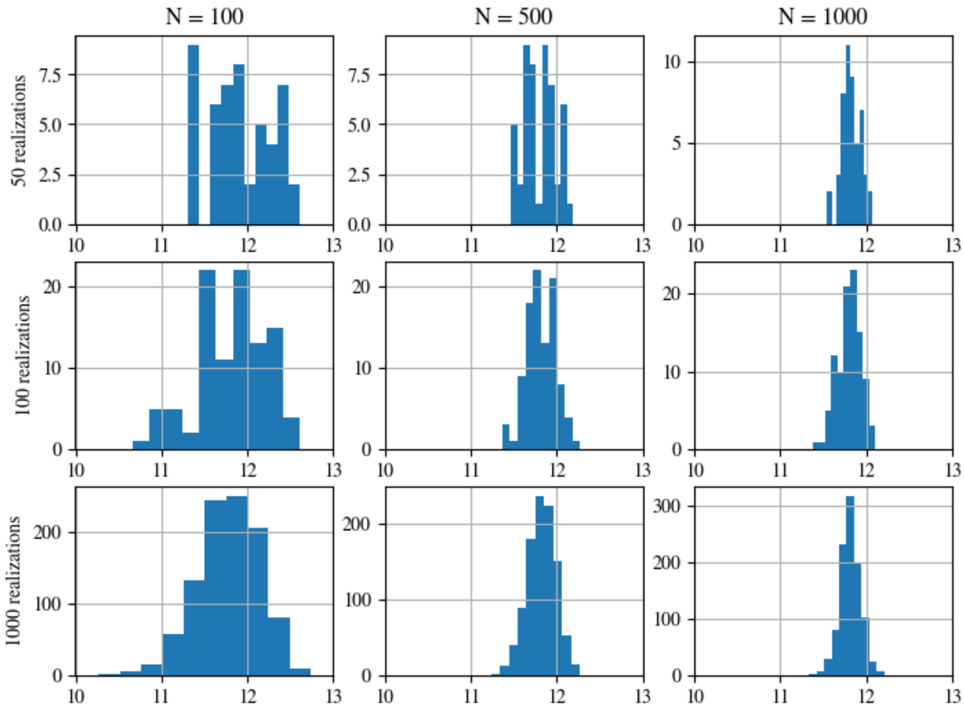
Berikutnya kita juga dapat mengamati distribusi dari statistik yang kita amati (dalam kasus ini adalah proporsi titik yang di bawah kurva dengan keseluruhan titik) dari beberapa realisasi atau percobaan yang independen satu sama lainnya.

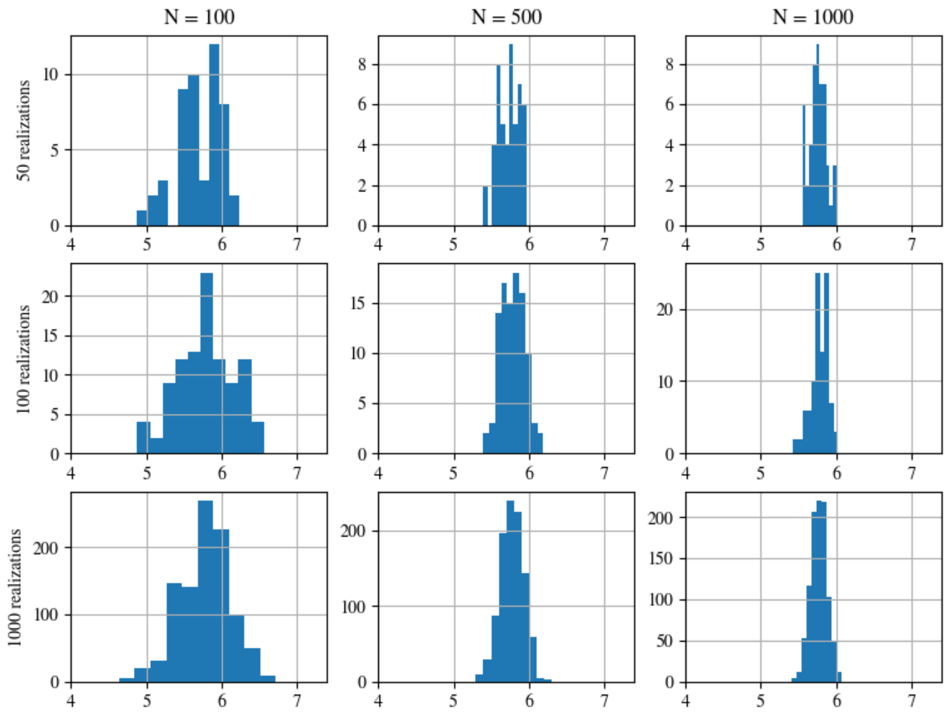
Berikut adalah distribusi nilai integral yang didapat untuk fungsi pertama jika menggunakan metode *direct sampling*. Baris pertama menunjukkan distribusi dari 50 realisasi independen, baris kedua 100, dan baris ketiga 1000, sedangkan satu kolom menyatakan jumlah titik sampel yang digunakan. Untuk kolom pertama menggunakan 100 titik sampel, kolom kedua 500 titik sampel, dan kolom ketiga 1000 titik sampel.



Dari gambar di atas terlihat bahwa semakin banyak realisasi/percobaan, distribusi dari nilai statistiknya semakin mendekati bentuk distribusi normal, dalam teori probabilitas hal ini dikenal sebagai ***Central Limit Theorem*** yang menyatakan bahwa bentuk distribusi nilai statistik yang diperoleh dari satu himpunan percobaan yang saling independen akan mendekati bentuk distribusi normal. Sedangkan dapat dilihat jika kita menggunakan jumlah sampel yang lebih banyak, variansi dari distribusi tersebut akan mengecil dan rata-rata empiris dari nilai statistiknya akan mendekati nilai ekspektasi atau nilai sebenarnya.

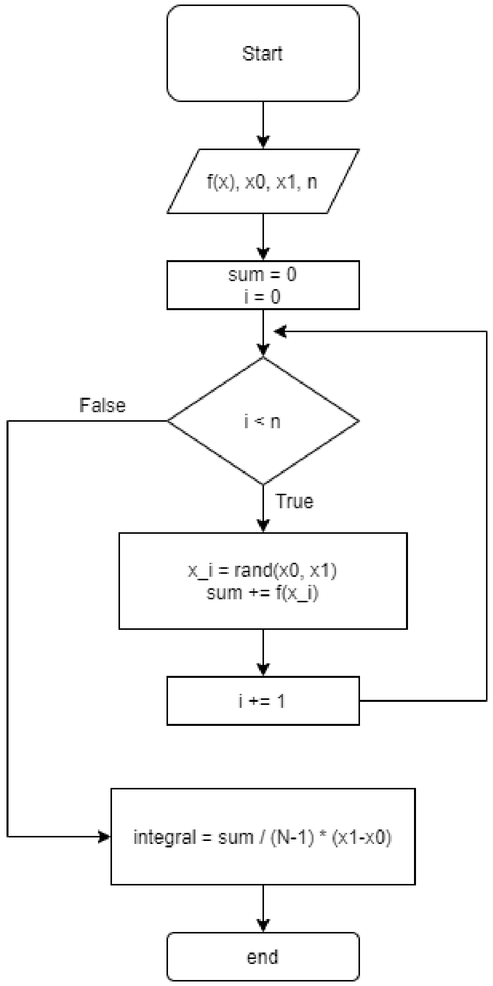
Berikut adalah plot distribusi untuk fungsi kedua dan ketiga





2. Monte Carlo Integration

2.1 Flowchart for Monte Carlo integration method

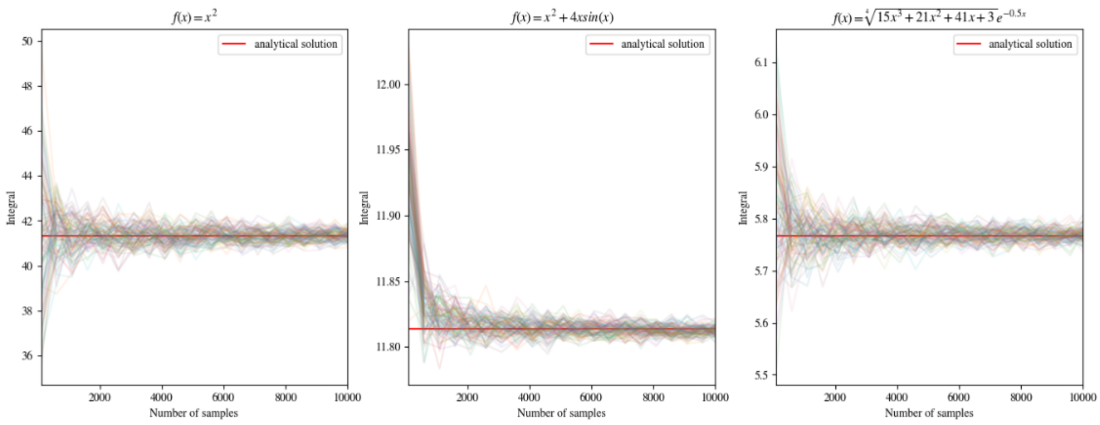


2.2 Code snippets

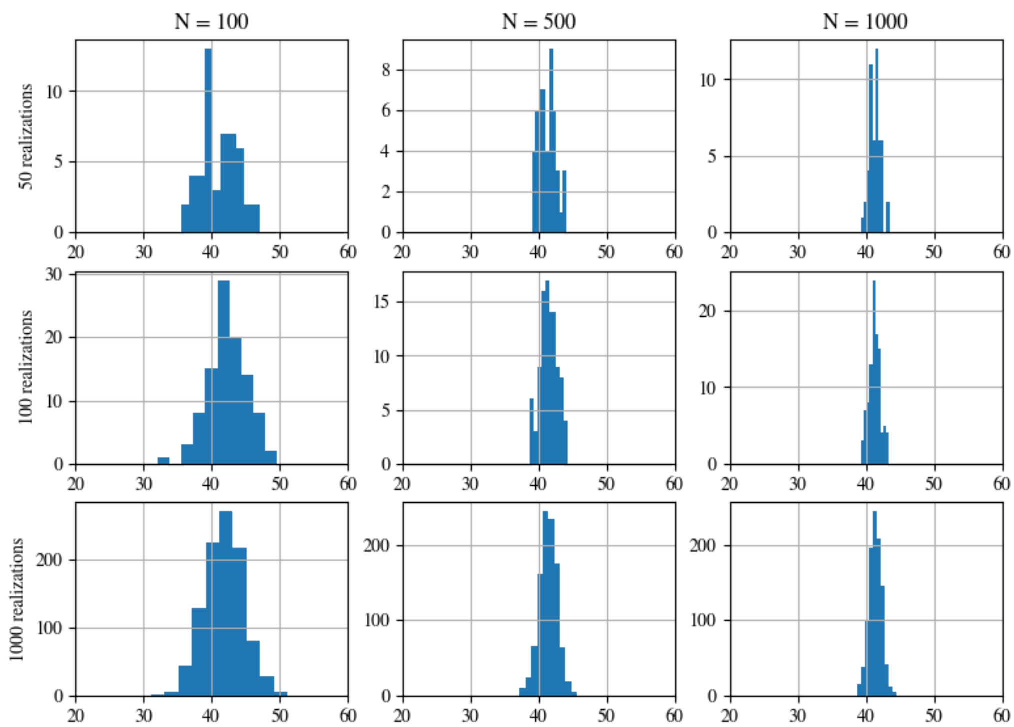
```python
def mc_integral(f, a, b, N):
    '''
    Perform Monte Carlo definite integration of a continuous function
    between interval a and b

    f : function of one variable
    a, b : lower and upper bound for integration
    N : sample size
    '''
    y_s = f(np.random.uniform(a, b, N))
    return np.sum(y_s) / (N-1) * (b-a)
```

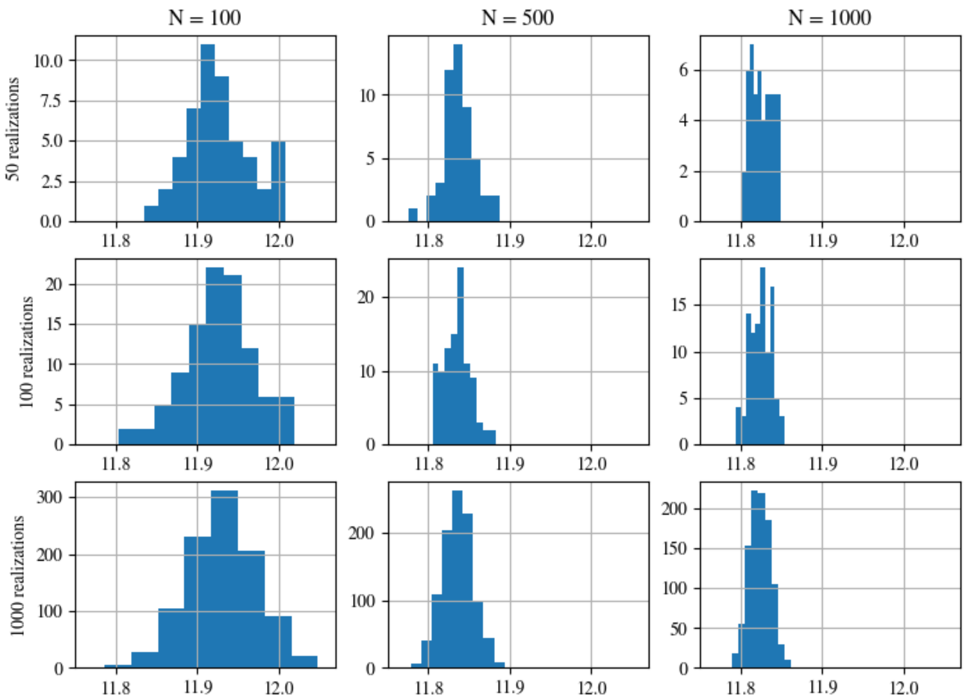
2.3 Analisis

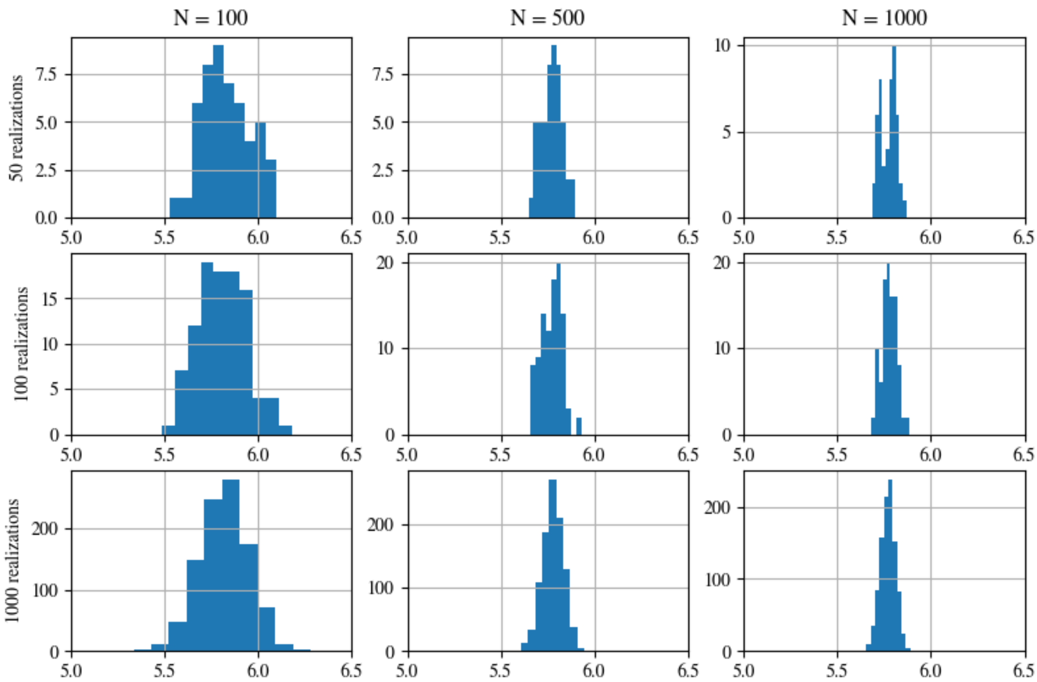
Berikut plot yang sama dengan plot di bagian sebelumnya, namun untuk ketiga fungsi. Dapat diamati juga peristiwa yang ditimbulkan oleh *Law of large number.*



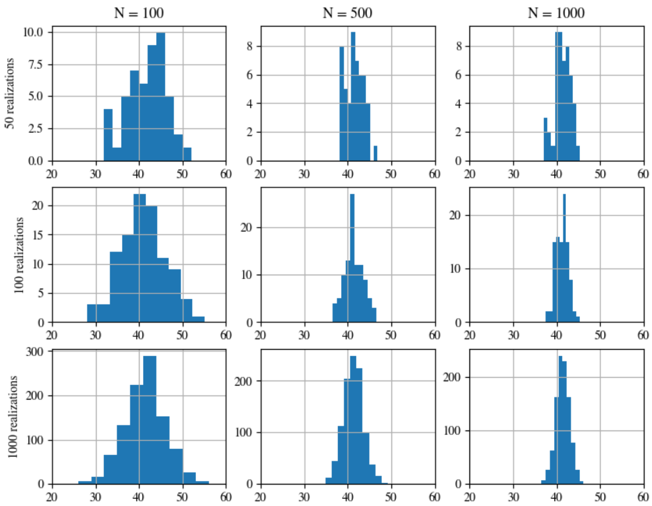
Berikut distribusi untuk fungsi pertama, kedua, dan ketiga



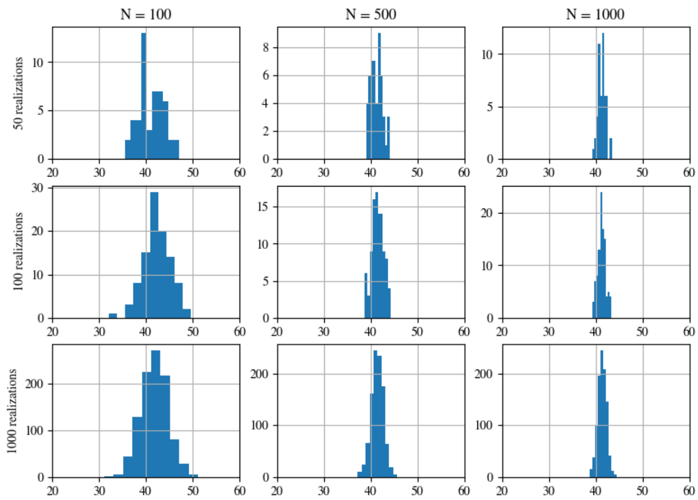

N = 100
N = 500
N = 1000
50 realizations
100 realizations
1000 realizations


N = 100
N = 500
N = 1000
50 realizations
100 realizations
1000 realizations

Jika disandingkan distribusi yang didapat dengan menggunakan metode direct sampling dan menggunakan perhitungan langsung terlihat bahwa pada jumlah sampel yang sama, variansi dari distribusi yang dihasilkan oleh metode integrasi Monte Carlo lebih kecil dibandingkan dengan hasil dari metode direct sampling.



Direct sampling



Monte Carlo integration

Metode kedua juga memberikan kelebihan yaitu kita tidak perlu menetapkan nilai maksimum untuk variabel random di sumbu y nya dan kita juga tidak perlu melakukan pengecekan apakah suatu titik yang dihasilkan secara acak terletak di bawah kurva, maka secara komputasi metode kedua bisa dikatakan lebih ringkas dibandingkan yang pertama.