Dasar Pemodelan dan Simulasi (CSH3H2) Minggu ke-9

Simulasi Monte Carlo

Program Studi Teknik Informatika, Universitas Telkom

January 10, 2019

Minggu ke-9

Aplikasi Metode MC: Menghitung Integral

Integral Numerik (Cara Lain)

Aplikasi Metode MC: Menghitung Integral, Minggu-9

Misal diberikan masalah menghitung integral:

$$I = \int_a^b f(x)\, dx \tag{1}$$

(a) Dengan metode quadrature untuk menghitung integral numerik:

$$I \approx \sum_{i=0}^{M} w_i f(x_i) \qquad M \text{ adalah jumlah partisi pada domain integral.}$$

Partisi rectangular: $w_i = \frac{b-a}{M}$, and $x_i = a + iw_i$
Partisi Trapezoidal: $w_i = \frac{(b-a)}{2M}, i = 1, 2, \cdots, M-1, w_0 = w_M = \frac{(b-a)}{M}$, and $x_i = a + iw_i$

Aplikasi Metode MC: Menghitung Integral cont

(b) Dengan metode MC, dilakukan dua tahap untuk mengestimasi nilai untuk I, yaitu menghitung mean dan variansi. Tahapan keseluruhan:

(i) Bangkitkan proses/variabel acak i.i.d: $x_i \sim U(a,b)$ sebanyak ukuran sampel N.

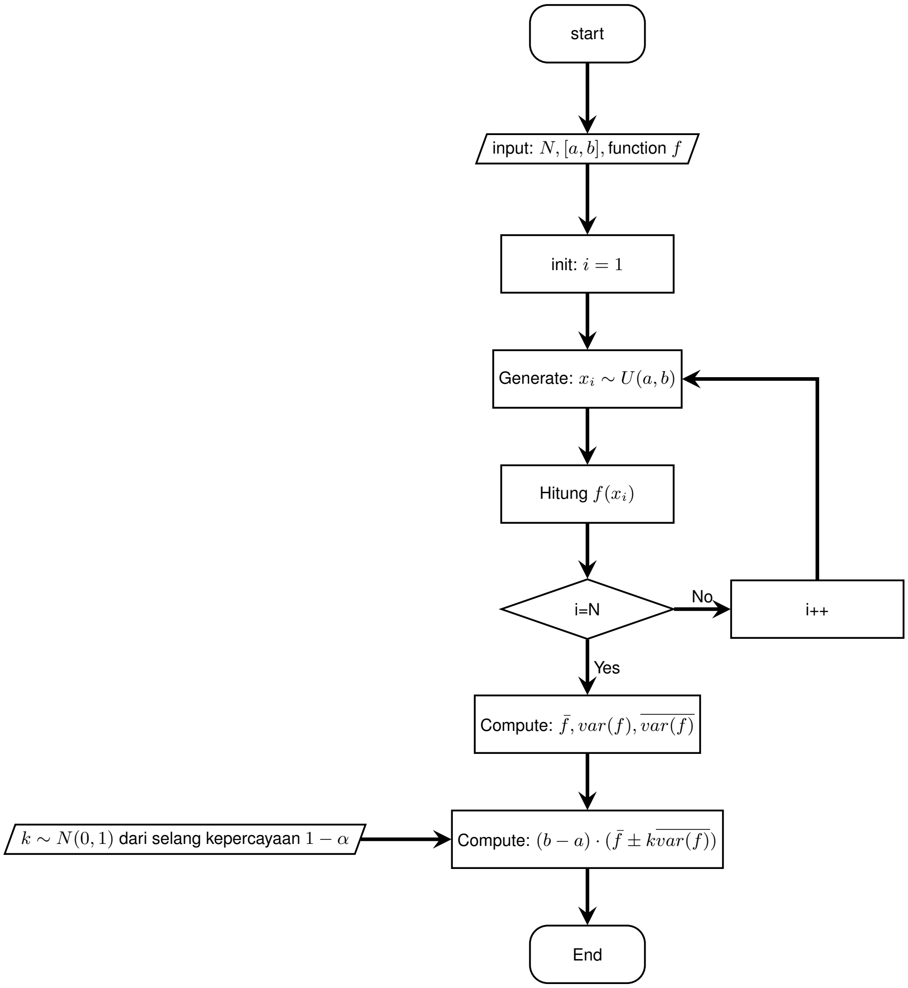
(ii) Hitung $f(x_i)$

(iii) Hitung nilai rata-rata $f(x_i)$: $\bar{f} = \frac{1}{N}\sum_{i=0}^{N} f(x_i)$, dan variansinya, $var(f) = \frac{1}{N-1}\sum_{i=0}^{N}(f(x_i) - \bar{f})^2 = \frac{1}{N-1}(f(x_i)^2 - N\bar{f}^2)$

(iv) Hitung $\overline{var(f)} = \frac{var(f)}{N}$, sehingga diperoleh selang kepercayaan untuk I: $I \in (b-a)(\bar{f} \pm k\,\overline{var(f)})$, nilai pengali $k \sim N(0,1)$ diperoleh dari selang kepercayaan tertentu.

Pada kasus integral diatas, volume integral $V = (b-a) = \int_a^b 1\,dx$. Secara umum pada kasus integral dimensi-$n > 1$, tidak selalu mudah membangkitkan bilangan acak uniform di sembarang V (next discussion!).

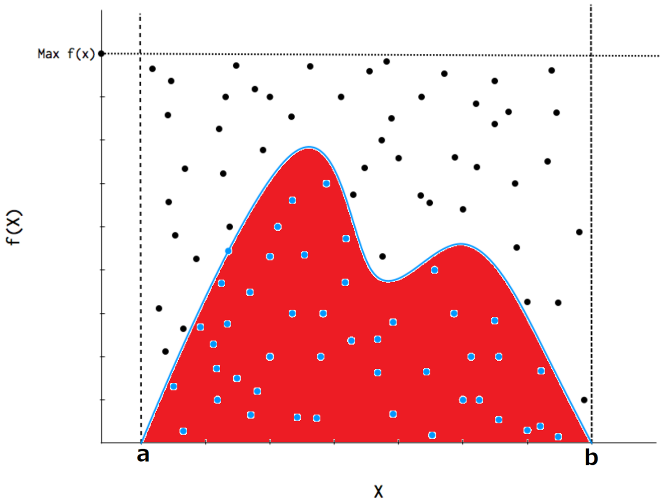
Algoritma I



start
input: $N, [a,b]$, function f
init: $i = 1$
Generate: $x_i \sim U(a,b)$
Hitung $f(x_i)$
i=N
No
i++
Yes
Compute: $\bar{f}, var(f), \overline{var(f)}$
$k \sim N(0,1)$ dari selang kepercayaan $1-\alpha$
Compute: $(b-a)\cdot(\bar{f} \pm k\overline{var(f)})$
End

Integral Numerik (Cara Lain simulasi MC)

Diberikan integral: $I = \int_{V=[a,b]} f(x)dx$, $x \in [a,b]$
Jika membangkitkan bilangan acak berdistribusi seragam di sembarang V tidak mudah, maka dapat dilakukan seperti menghitung luas area.



Definisikan fungsi $g(w_i) = (x_i, y_i) \in [a,b] \times [\min f, \max f]$:

$$g(w_i) = \begin{cases} 1 & \text{jika } y_i \leq f(x_i) \\ 0 & \text{lainnya} \end{cases} \tag{2}$$

$$I = \frac{\sum_{i=1}^{N} g(w_i)}{N} (\max f - \min f) \cdot (b-a) \tag{3}$$

$$= \frac{\text{\# titik di area merah}}{\text{\# Total sampel titik}} (\max f - \min f) \cdot (b-a) \tag{4}$$

Membangun algoritma II integral numerik

0: Input: fungsi f, selang $[a, b]$, N, dan inisiasi $i = 1$.

(i) Tentukan $\min_{x\in[a,b]} f(x)$ dan $\max_{x\in[a,b]} f(x)$ pada selang $[a, b]$.

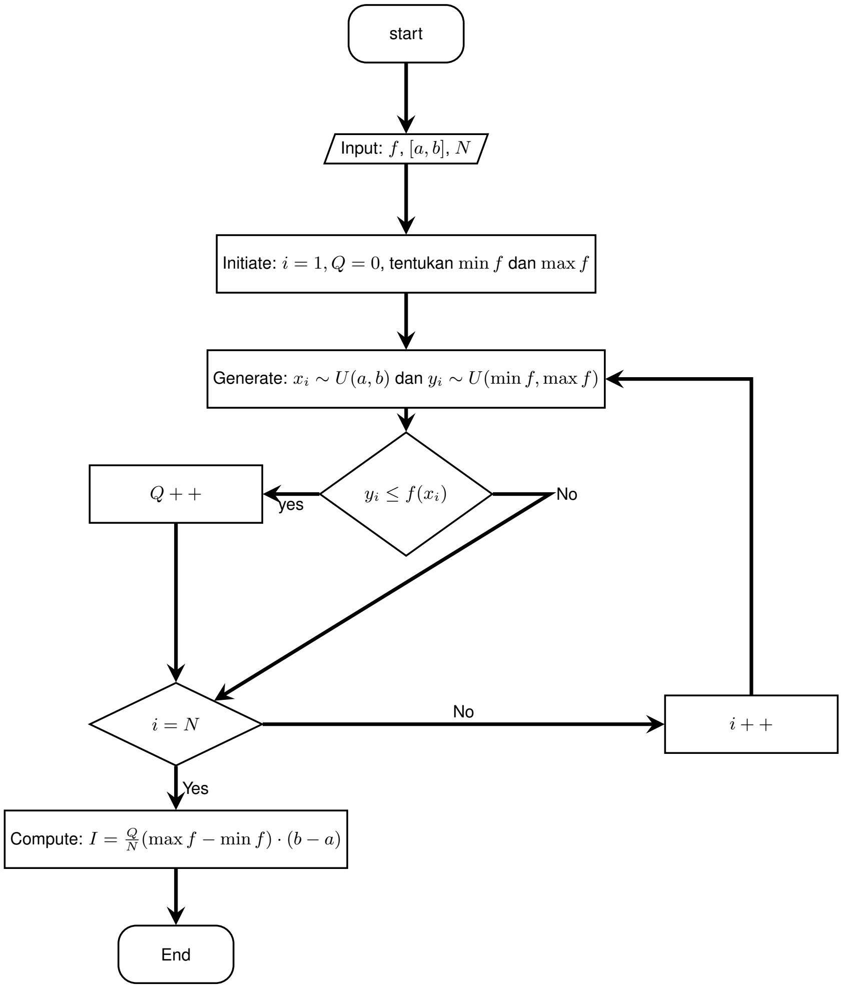
(ii) Bangkitkan pasangan bilangan acak $w_i = (x_i, y_i)$, dimana $x_i \sim U(a, b)$ dan $y_i \sim U(\min f, \max f)$.

(iii) Hitung $g(w_i)$.

(iv) Jika $i < N$, $i = i + 1$ dan kembali ke langkah (ii).

(v) Jika $i = N$, hitung I dengan (3). Selesai.

Flow chart Algoritma II



Program python metode rectangular & trapezoidal

python's libraries & Contoh fungsi

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

def easy_function(x):
    return((3)*(x**2))

def hard_function(x):
    return((1/np.sqrt(2*np.pi))*np.exp(-(x**2)/2))

X=np.linspace(-5,5,1000)
plt.plot(X,easy_function(X))
plt.xlim=((-5,5))
plt.show()
plt.plot(X,hard_function(X))
plt.ylim=((-1,1))
plt.show()
```

program integral numerik Rectangular & Trapezoidal

```python
def integrate_rec(x1,x2,func=easy_function,n=10000):
    dx=(x2-x1)/n
    Xs=np.linspace(x1,x2,n+1)
    return np.sum(func(Xs))*dx

def integrate_trapez(x1,x2,func=easy_function,n=10000):
    return integrate_rec(x1,x2,func,n=10000)-0.5*(x1+x2)/n
```

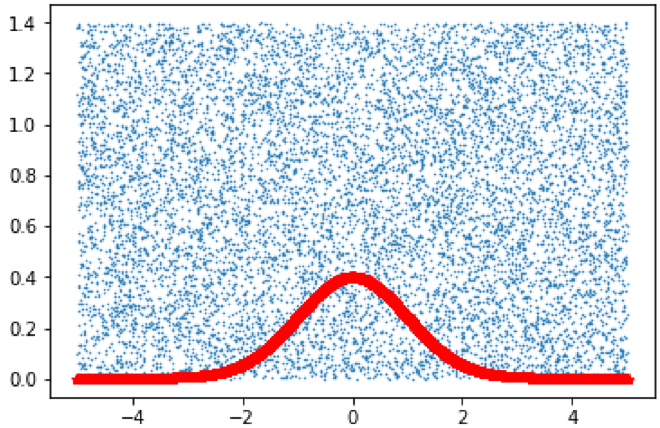
Program python algoritma I

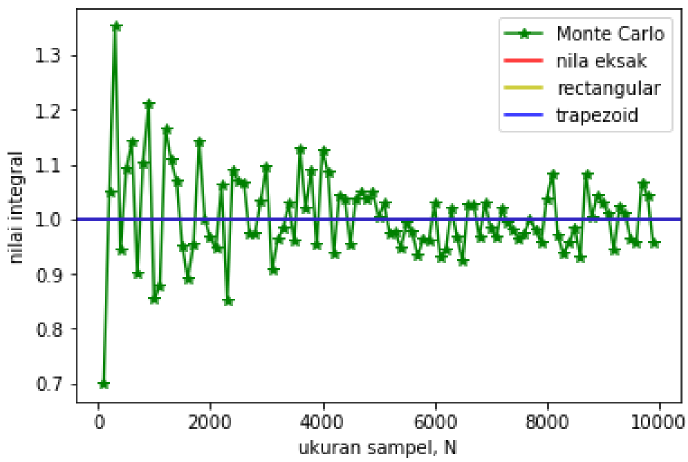
program integral numerik dengan simulasi Mote carlo

```
def integrate_alg1(x1,x2,func=easy_function,n=10000):
    X=np.random.uniform(x1,x2,n)
    plt.scatter(X,func(X))
    plt.show()
    return np.mean(func(X))*(x2-x1)

def integrate_alg2(x1,x2,func=easy_function,n=10000):
    X=np.linspace(x1,x2,1000)
    y1=min(func(X))
    y2=max((func(X)))+1
    print(x1,x2,y1,y2)
    area=(x2-x1)*(y2-y1)
    x=np.random.uniform(x1,x2,n)
    y=np.random.uniform(y1,y2,n)
    check=[1 if b<=func(a) else 0 for a,b in zip(x,y) ]
    plt.scatter(x,y, s=0.2)
    plt.plot(x,func(x),'*r')
    plt.show()
    return np.sum(check)*area/n


print(integrate_rec(-5,5,func=hard_function))
print(integrate_trapez(-5,5,func=hard_function))
print(integrate_alg1(-5,5,func=hard_function))
print(integrate_alg2(-5,5,func=hard_function))
```

Hasil Simulasi dan Perbandingan

$$\int_{-5}^{5} \frac{1}{\sqrt{2\pi}} \exp^{-\frac{x^2}{2}} \; dx$$





$$\int_{-5}^{5} 3x^2 \, dx$$