Monte-Carlo Integration

(Probabilistic Integration Method)

Monte Carlo Method

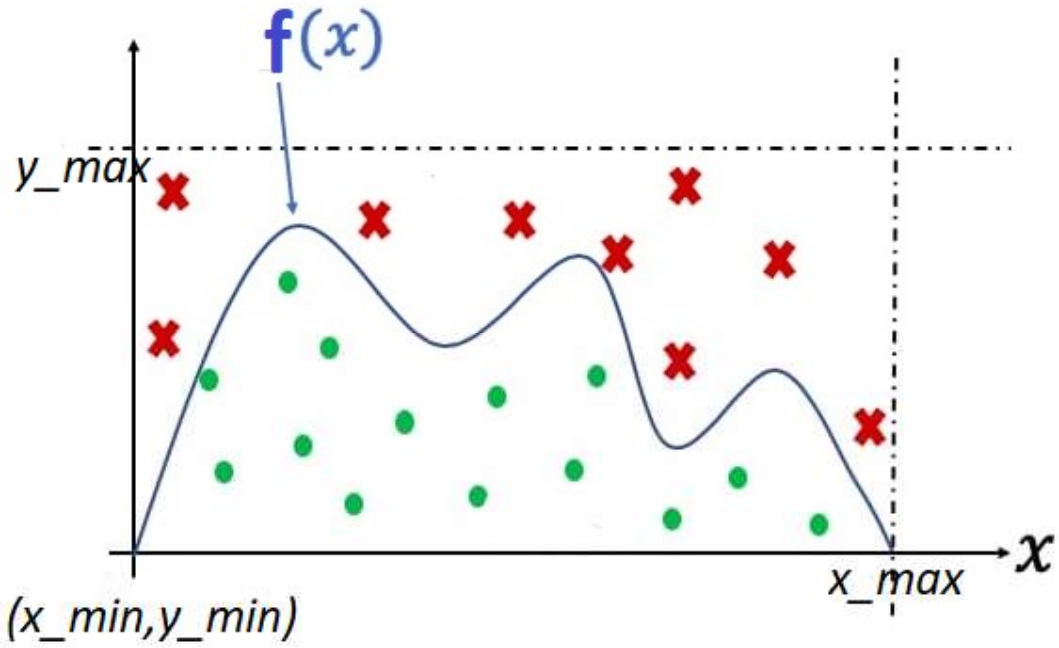
Monte Carlo Methods (M C Methods) are abroad range of computational algorithms that rely on repeated random sampling to make numerical estimation of an unknown parameter .

In general ,the method follows these steps

- **Define a domain of possible inputs**
- **Generate random samples under that domain**
- **Deterministic computation**
- **Analyze to get certain numerical results**

Monte-Carlo Integration:Technique

- Set the target Rectangular area with limit (x_min,x_max) and (y_min,y_max).
- Throw N darts in the target area.
- Compute the no of darts falling under the rectangular curve.



- $$\int_a^b ydx = \frac{Inside\ Points}{Total\ Points}\ X\ Area\ of\ the\ Rectangle$$

Monte-Carlo Integration:Python Code

```
import random
import numpy as np
from scipy.integrate import simps
import matplotlib.pyplot as plt

def f(x):
   return np.sin(x)**2

a,n=0,100
b=np.pi
x=np.linspace(a,b,n)
y=[f(i) for i in x]

y_min=np.min(y)
y_max=np.max(y)*1.02
rect_area=(b-a)*(y_max-y_min)
```

Monte-Carlo Integration:Python Code

```
x_up=[ ]
x_dn=[ ]
y_up=[ ]
y_dn=[ ]

trial=1000
count=0
for i in range(trial):
    x_trial=a+(b-a)*random.random()
    y_trial=y_min+(y_max-y_min)*random.random()

    if y_trial<=f(x_trial):
        count=count+1
        y_dn.append(y_trial)
        x_dn.append(x_trial)
    else:
        y_up.append(y_trial)
        x_up.append(x_trial)
```

Monte-Carlo Integration:Python Code

```
integration=rect_area*count/trial
print("Integration=",integration)
acc_int= simps(f(x),x)
print("Actual Integration=",acc_int)


plt.plot(x,y,color='m',lw=5)
plt.plot(x_up,y_up,'.',color='g')
plt.plot(x_dn,y_dn,'.',color='r')
plt.axhline(y=y_max,color='k')
plt.axhline(y=y_min,color='k')
plt.axvline(x=a,color='k')
plt.axvline(x=b,color='k')
plt.show()
```

Monte-Carlo Integration:Output

Integration= 1.563365516021061

Actual Integration= 1.5708016509046652