task3

October 6, 2023

[5]:
```
import numpy as np
```

[6]:
```
# implement the integrand f(x) as a Python function
def f(x):
    f_integrand = x # the integrand is f(x)= x
    return f_integrand
```

[7]:
```
N=1000
```

[8]:
```
x_i = np.random.uniform(size=N)
```

[9]:
```
print( "Shape of x_i is",x_i.shape)
```

```
Shape of x_i is (1000,)
```

[10]:
```
f_i = f(x_i) # evaluate the integrand at these N samples
```

[11]:
```
print( "Shape of f_i is" , f_i .shape)
```

```
Shape of f_i is (1000,)
```

[12]:
```
I_N = np .sum( f_i ) / N
```

[13]:
```
print( "Monte Carlo approximation is %f "%I_N)
```

```
Monte Carlo approximation is 0.501743
```

[ ]:
```

```