Assignment_7

November 1, 2023

1 Computational Methods in Stochastics - Assignment 7

[28]:
```
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm
```

1.0.1 a)

We know that

$U(q) = \lambda e^{-\lambda q} = -\log[e^{-\lambda q}] = -\log\{\exp[-\lambda q + \log(\lambda)] = \lambda q - \log\lambda$

So we would have:

$\Rightarrow \forall q \nabla U = dU(q)/dq = \lambda$

1.0.2 b)

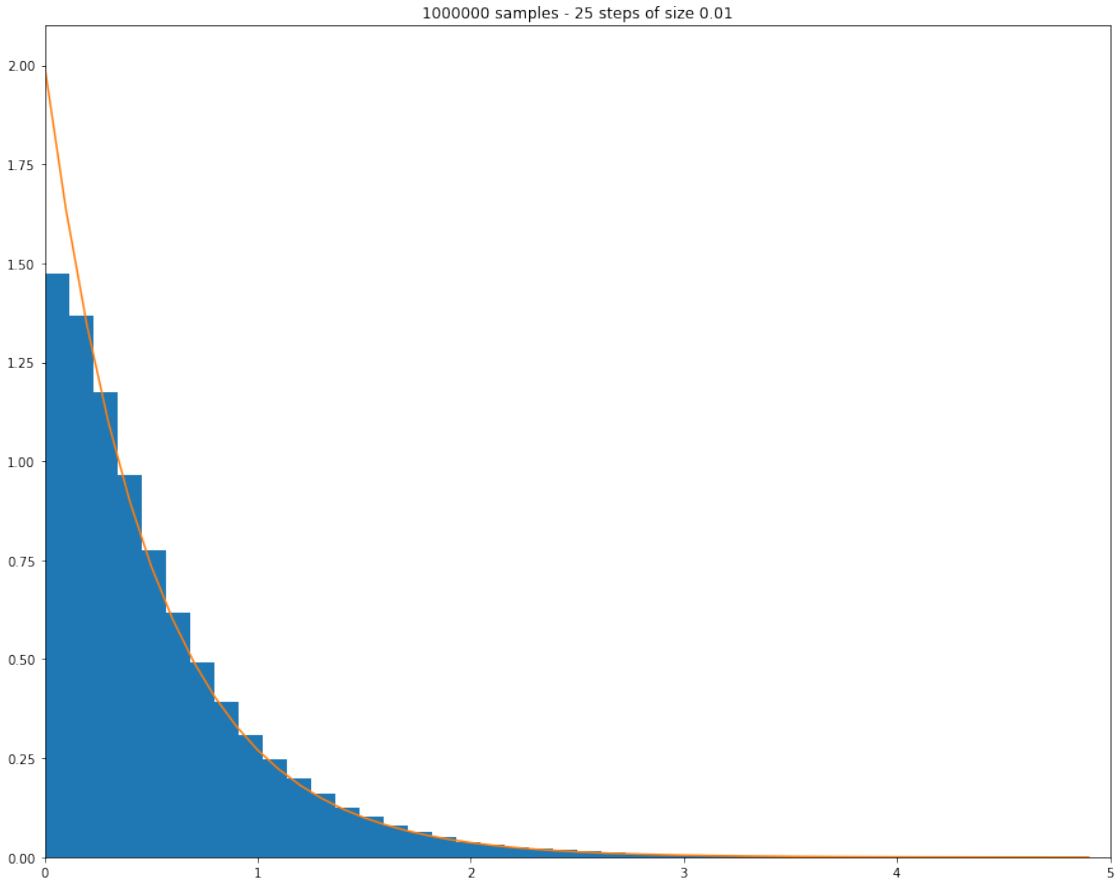
Functions needed in the HMC Algorithm: - **Exponential Function**: $Exp(x, l) = l \times e^{-l \times x}$ - **Potential Energy Function**: $U(q) = -log\lambda + \lambda q$ - **Energy Function**: $H(q, p) = K(p) + U(q)$ - **Derivative of Potential Energy Function**: $\frac{d}{dq}[U(q)] = \lambda$

[29]:
```
Lambda = 2
N = 1000000
L = 25
BINS = 50

def Exponential(x, l):
    return l*np.exp(-l*x)

def U(q):
    return -np.log(Exponential(q, Lambda))

def grad_U(q):
    return Lambda
```

1.0.3 c and d)

[30]:
```
EPSILON = 0.01
q0 = np.random.random()
samples = []

def Update_p_q(q, p):
    for iter in range(1,L+1):
        q = q + EPSILON*p
        if(iter != L):
            p = p - EPSILON * grad_U(q)
    return q, p

def Calc_p(p, q):
    return p - EPSILON * grad_U(q)/2

def U_computations(q0, q1):
    return U(q0), U(q1), U(q0) - U(q1)

def K_computations(p1, p0):
    return p1**2/2, p0**2/2, p1**2/2 - p0**2/2

for i in tqdm(range(N)):
    q1 = q0
    p0 = np.random.normal()
    p1 = p0
    p0 = Calc_p(p0, q1)
    q1, p0 = Update_p_q(q1, p0)
    p0 = -Calc_p(p0, q1)
    U_current, U_prop, diff_U  = U_computations(q0, q1)
    K_current, K_prop, diff_K = K_computations(p1, p0)

    if (np.random.random() < np.exp(diff_U + diff_K)): # constraint 1
        q0 = q1
    if (q0 < 0): # constraint 2
        q0 = -q0
    samples.append(q0)

plt.figure(figsize=(15, 12))
plt.hist(samples, density=True, bins=BINS)
plt.plot(np.arange(0,5,0.1), Exponential(np.arange(0,5,0.1), Lambda))
plt.xlim(0,5)
plt.title(str(N)+" samples - " + str(L) + " steps of size " + str(EPSILON))
plt.show()
```

```
100%|      | 1000000/1000000 [00:14<00:00, 68664.81it/s]
```


1000000 samples - 25 steps of size 0.01
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5