q32

April 19, 2020

[ ]:

```
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:

```
def h(t,a,b,c,d):
    return a*np.exp(b*t) + c*np.exp(d*t)
```

[ ]:

```
def momentum_update(x0, A, eta, gamma, num_iters=200):
    X = [x0*1.0]
    x = x0*1.0
    g = 0.0
    for t in range(num_iters):
        g = (1-gamma)*g + gamma*(2*A*x)
        x = x - eta*g
#         print(x)
        X.append(x*1.0)
    return X
```

[ ]:

```
a = 1.270711
b = -0.036617
c = -0.207107
d = -0.213388
x0 = 1.0
A = 1.0
eta = 1/64
gamma = 1/4
num_iters = 200
def h(t):
    return a*np.exp(b*t) + c*np.exp(d*t)

t = np.arange(0,200,0.01)
H = h(t)
X = momentum_update(x0, A, eta, gamma, num_iters)
plt.plot(t,H,label=r'continuous estimate of $x_t$')
plt.plot(X, label=r'True $x_t$')
plt.xlabel("t")
plt.ylabel(r'x_t')
plt.title(r'$x_0 = {}, a = {}, \eta = 1/{}, \gamma = 1/{}$'.format(x0,A,64, 4))
plt.legend()
plt.savefig("./q32_1.png")
plt.show()
```

[ ]:

```
a = 1.270711
b = -0.036617
c = -0.207107
d = -0.213388
x0 = 1.0
A = 1.0
eta = 1/64
gamma = 1/8
num_iters = 200
def h(t):
    return (1/16)*(np.exp(-t/16))*(t+16)

t = np.arange(0,200,0.01)
H = h(t)
X = momentum_update(x0, A, eta, gamma, num_iters)
plt.plot(t,H,label=r'continuous estimate of $x_t$')
plt.plot(X, label=r'True $x_t$')
plt.xlabel("t")
plt.ylabel(r'x_t')
plt.title(r'$x_0 = {}, a = {}, \eta = 1/{}, \gamma = 1/{}$'.format(x0,A,64, 8))
plt.legend()
plt.savefig("./q32_2.png")
plt.show()
```

[ ]:

```
a = 1.270711
b = -0.036617
c = -0.207107
d = -0.213388
x0 = 1.0
A = 1.0
eta = 1/64
gamma = 1/16
num_iters = 200
def h(t):
    return np.exp(-t/32)*(np.sin(t/32) + np.cos(t/32))

t = np.arange(0,200,0.01)
H = h(t)
X = momentum_update(x0, A, eta, gamma, num_iters)
plt.plot(t,H,label=r'continuous estimate of $x_t$')
plt.plot(X, label=r'True $x_t$')
plt.xlabel("t")
plt.ylabel(r'x_t')
plt.title(r'$x_0 = {}, a = {}, \eta = 1/{}, \gamma = 1/{}$'.format(x0,A,64, 16))
plt.legend()
plt.savefig("./q32_3.png")
plt.show()
```

[ ]:

```
a = 1.270711
b = -0.036617
c = -0.207107
d = -0.213388
x0 = 1.0
A = 901
eta = 1/64
gamma = 1/16
num_iters = 200
def h(t):
    return (1/30)*np.exp(-t/16)*(np.sin(15*t/8) + 30*np.cos(15*t/8))

t = np.arange(0,200,0.01)
H = h(t)
X = momentum_update(x0, A, eta, gamma, num_iters)
plt.plot(t,H,label=r'continuous estimate of $x_t$')
plt.plot(X, label=r'True $x_t$')
plt.xlabel("t")
plt.ylabel(r'x_t')
plt.title(r'$x_0 = {}, a = {}, \eta = 1/{}, \gamma = 1/{}$'.format(x0,A,64, 8))
plt.legend()
plt.savefig("./q32_4.png")
plt.show()
```