```python
##############################################################################
###Code for figure 4.1
###Generating m paths of the geometric brownian motion (GBM)
###on the time intervall [0,T]
###SDE_SamplingGBM.py
###Python 2.7
##############################################################################

import numpy as np
import numpy.matlib
import matplotlib.pyplot as plt
from NumericalSDE import *

#################################################
###Geometric brownian motion (SDE)
### dXt = a(Xt)dt + b(Xt)dWt
### X0 = x0
### a(x) = mu*x, b(x) = sigma*x
### mu, sigma constants
### True solution:
###
#################################################
#Parameter
sigma = 1.5
mu = 1.0
#starting value x0
x0 = 1
#Parameters for the discretization
n =2**8
t = timegrid(n)
#m discretized Wiener processes
m = 5
w = np.zeros((n+1,m))
for k in range(0,m):
    w[:,k] = wiener(n)
#m sample paths
Xt = np.zeros((n+1,m))
for k in range(0,m):
    Xt[0,k] = x0
    for j in range(0,n):
        Xt[j+1,k] = Xt[0,k]*np.exp((mu-sigma**2/2)*(t[k+1]) + sigma*w[j+1,k])


#Plot
for sample_path in Xt.T:
    plt.plot(t, sample_path,'r',linewidth=0.5)
plt.xlabel('t', fontsize=16)
plt.ylabel('x', fontsize=16)
plt.show()
```