```
###########################################################################
###Code for figure 2.2
###Plots a discretized Wiener process and finer versions of the same process
###using the refinement algorithm (see module NumericalSDE)
###WienerRefinementGraphic.py
###Python 2.7
###########################################################################
import numpy as np
import matplotlib.pylab as plt
from NumericalSDE import *

#Number of steps.
n = 16
#Create an empty array to store the realizations
en = n
x = wiener(en)
y = refineWiener(x)
z = refineWiener(y)

#timegrids
t_1 = timegrid(en)
t_2 = timegrid(2*en)
t_3 = timegrid(4*en)

#plot
plt.plot(t_1, x,'k', linewidth=0.8)
plt.plot(t_2, y,'r', linewidth=0.5, c='b')
plt.plot(t_3, z,'r', linewidth=0.5, c='b')

plt.xlabel('t', fontsize=16)
plt.ylabel('x', fontsize=16)
plt.show()


```