*5. Use Python to generate a wav file of a sine wave at 8 bits per sample. Read your 8 bit/sample wav file into MATLAB and plot the signal to verify that it is a sine wave (zoom in if necessary to show the waveform.) Verify that* *the quantization step size is as expected and verify its spectrum. Is there any noticeable effect of a lower number of bits/sample on the sound quality (keeping the same number of samples/second)? Submit your Python code,* *wav file, and* *written comments.*

**wav file:** sin\_8bits\_mono.wav

**Python code：**sin\_8bits.py

from struct import pack

from math import sin, pi

import wave

Fs = 8000

wf = wave.open('sin\_8bits\_mono.wav','w')

wf.setnchannels(1)#mono

wf.setsampwidth(1)#8 bits per sample - 1 bytes

wf.setframerate(Fs)#frequency I choosed is 8000Hz

maxAmp = 2\*\*7 - 1.0#8 bits - 0~7

f = 261.625565

for n in range(0, int(0.5\*Fs)):

#'B'-unsigned 8-bit wav and it standard size is 1

binary\_string = pack('B', maxAmp \* sin(n\*2\*pi\*f/Fs)+128)

wf.writeframesraw(binary\_string)

wf.close()

**written comments：**the quantization step size is 1/svp as expected and equal to 128, spectrum show that the highest is 261.7Hz , which is the frequency I made. lower number of bits/sample just make the sound quality decline. Since sound wave can only have unsigned 8-bit( use ‘B’), so we need to add 128 to the output that make sure it have a positive output.

Plus, why do I have a wrong quantization step size if I change the f = 261.625565, to f = 200, or f = 600. All these make qss become no more 128. From the pack we can get that f is the frequency we have in the wave. So why it could cause the huge change of svp?