2. Magnitude and phase spectra
The code function is shown at the last page.

Part 2a,2b:
windowStart=2048, windowSize=256, threshold = 0.1

windowStart=2048, windowSize=1024, threshold = 0.1
windowStart=2048, windowSize=512, threshold = 0.1

windowStart=2048, windowSize=512, threshold = 1.0
Part A. With the increase of the window size, the frequency resolution increases. Since this is the speech signal, with bigger window sizes (like 256, 512 or 1024 samples) we get STFT of voiced speech that consists of a set of narrow “harmonic lines”, which are shaped by the magnitude of the product of the glottal flow Fourier transform and vocal tract transfer function.

Part B. The more we increase the threshold, the more magnitude peaks we remove (marked as “+” at the graphs). Thus we loss the quality of the re-synthesis (see part C and part 3)

Part C:
windowStart=2048, windowSize=256, threshold = 0.1
At the short window (256 samples), the influence of a relatively small threshold is big, since the ratio of the number of peaks removed to the total number of peaks in the window is high. Therefore the distortion rate is high. Within a long window the number of removed peaks is higher, however the total number of peaks is much higher than in the case of short window. Therefore the percentage of removed peaks is smaller, thus the distortion. Long window have also much more linear phase after unwrapping.

3. **Influence of cutting the peaks on the sound.**

3a) Window length of 128 samples is too short, you can hear “metallic” features of the speech. Window length 1024 samples is too long, you can hear some distortions sounding like echo or delicate reverberation. With window sizes 256 and 512 samples the distortions are barely audible.

3b) With threshold 0.1 the distortions are minimal, but quickly increase with the growth of the threshold. The sound seems to be “underwater”. With the threshold = 5.0 you can barely recognize the speech.

3c) The best sound can be obtained using t=0.1 and window size = 512 samples. However, the transformations still cut some some peaks, even with t=0.1. With t=0 (no peak cutting, an extra experiment) the sound is not distorted, however in every case the sound is audibly quieter (the magnitude is couple of times smaller) than the original sound.
The code:

function y = sinemodelSingle (x, windowStart, windowSize, threshold)
    % y = sinemodel (x, windowSize, threshold)
    % function for sound analysis and synthesis based on spectral peaks
    % single window
    % x: input sound
    % windowStart: start of the window
    % windowSize: size of analysis window, has to be the power of 2
    % threshold: magnitude threshold to be used
    % y: synthesized sound
    window = hanning(windowSize); % analysis window
    N = windowSize*2; % size of FFT
    pin = windowStart; % pointer for input sound
    pend = windowStart+windowSize;
    if pend>length(x),
        'Window exceeds the sample size, stripping'
        pend = length(x);
    end;
    %hopSize = windowSize/2;
    fftframe = zeros(N,1);
    y = zeros(windowSize,1);
    %hold on;
    %--- windowing
    inframe = x(pin+1:pin+windowSize).* window(1:windowSize);
    figure(1);
    size(inframe)
    %--- circular shift for zero-phase windowing
    fftframe(1:windowSize/2) = inframe(windowSize/2+1:windowSize);
    fftframe(N-windowSize/2+1:N) = inframe(1:windowSize/2);
    %--- fft computation
    spectrum = fft(fftframe); % complex spectrum
    magspec = abs(spectrum); % magnitude spectrum
    phasespec = angle(spectrum); % phase spectrum
    uphasespec = unwrap(phasespec); % unwrapped phase spectrum
    %-- peak detection
    specder = diff([threshold; magspec(:); threshold]); % derivate of magnitude spectrum
    ploc = find(magspec(1:N) >= threshold & specder(1:N)>0 & specder(2:N+1) <= 0); % peak locations
    pmag = magspec(ploc); % peak magnitudes
    pphase = phasespec(ploc); % peak phases
    upphase = uphasespec(ploc); % peaks of unwrapped phases
    xLen=1:length(magspec);
    thPlot = threshold*ones(1,length(magspec));
    threshold %print threshold
    %subplot(2,1,1);
    %plot(xLen,magspec,'b',ploc,pmag,'b+',xLen,thPlot,'r:');
    %subplot(2,1,2);
    %plot(1:length(phasespec),phasesspec,'r',ploc,pphase,'r+');
%--- synthesize sound from peaks

outspectrum = zeros(N,1);
outspectrum(ploc) = pmag.*exp(i.*pphase); % fill positive frequencies
outspectrum(N-ploc+1) = pmag.*exp(i.*(-pphase)); % fill negative frequencies
sframe = real(ifft(outspectrum)); % compute inverse fft
sframe1 = [sframe(N-windowSize/2+1:N); sframe(1:windowSize/2)]; % undo circular shift
y = sframe1 .* window; % overlapp-add and apply synthesis window

%pin = pin + hopSize;
%hold off
%figure
subplot(4,2,1);
plot(inframe);
title('inframe');
subplot(4,2,2);
plot(fftframe);
title('fftframe');
subplot(4,2,3);
plot(xLen,magspec,'b',ploc,pmag,'b+',xLen,thPlot,'r:');
title('magnitude spectrum');
subplot(4,2,4);
plot(xLen,uphasespec,'b',ploc,upphase,'r+');
title('phase spectrum');
subplot(4,2,5);
plot(abs(outspectrum))
title('output magnitude spectrum');
subplot(4,2,6);
plot(sframe);
title('sframe');
subplot(4,2,7);
plot(sframe1);
title('sframe1');
subplot(4,2,8);
plot(y);
title('y');