Audio Content Analysis
Assignment # 1
Instructor: Juan Pablo Bello

A Implementation [7 pts]

In this section you will be implementing a series of functions, and finally a script tying them all together. Each function should be saved as a separate .m file, where the filename is identical to the function name. The functions definitions should ***exactly*** *follow the given code. Submit all code to NYU classes as a single zip file named "YourLastName1.zip".*

1. Write a Matlab function that creates a sine sweep using the equation

$$x(t) = a \cdot \sin\left(\texttt{cumsum}\left(\frac{2\pi\mathbf{f}}{f_s}\right)\right)$$

where a is the amplitude, f_s is the sampling frequency and $\mathbf{f}$ is a vector of logarithmically spaced frequency values. Use Matlab's `logspace` function for this. The `cumsum` Matlab function is a running sum operator which considers the cumulative effect of the frequency changes in the current sample phase [2 pts].

```
function x_t = sine_sweep(f_min, f_max, fs, duration, a)
%    Generate a sine sweep.
%
%    Parameters
%    ----------
%    f_min : float
%        minimum frequency (Hz)
%    f_max : float
%        maximum frequency (Hz)
%    fs : int
%        sample rate (samples per second)
%    duration : float
%        length of sweep (seconds)
%    a : float
%        amplitude
%
%    Returns
%    -------
%    x_t : 1 x T array
%        time domain signal
```

```
>>  %% Example call %%
>>  fs = 44100;
>>  x_t = sine_sweep(500.0, 5000.0, fs, 1.0, 0.95);
>>  soundsc(x_t, fs); % plays a time series x_t at sample rate fs
```

2. Write a Matlab function that calculates and plots the spectrogram of a time domain audio signal, such that the x-axis represents time, the y-axis represents frequency, and the z-axis (the figure's colorcode) represents the spectral magnitude in dB. The point $(0, 0)$ should be in the bottom left corner of the plot, and all axes should be appropriately labeled and scaled. Use Matlab's `spectrogram` and `imagesc` functions for this (read their help files carefully before using them) [2 pts].

```
function [] = plot_spectrogram(x_t, win_size, hop_size, win_type, fs, nfft)
%    Calculate and plot the spectrogram of a time-domain audio signal.
%
%    Parameters
%    ----------
%    x_t : 1 x T array
%        time domain signal
%    win_size : int
%        window size (in samples)
%    hop_size : int
%        hop size (in samples)
%    win_type : str
%        window type (one of \texttt{`rect', `hamm', `black'})
%    fs : int
%        sample rate (samples per second)
%    nfft : int
%        fft length (in samples)
%
%    Returns
%    -------
%    None
```

```
>> %% Example Call %%
>> N = 128;
>> h = N/2;
>> fs = 44100;
>> x_t = sine_sweep(500.0, 5000.0, fs, 1.0, 0.95);
>> plot_spectrogram(x_t, N, h, 'rect', fs, N); % plots spectrogram
```

3. Implement a Matlab function for the computation of the spectrogram using matrix multiplication (as discussed in class) rather than the `fft` function. Use the `buffer` function to break the signal into N-long overlapping blocks. Make sure this function operates exactly as the `spectrogram` function [2pts].

   ```
   function [S, F, T] = my_spectrogram(x, window, noverlap, nfft, fs)
   ```

   Where each variable is consistent with Matlab's `spectrogram` function.

4. Write a Matlab script called `assignment1.m` which runs the code used to generate the plots for your report (see the Analysis section below) [1 pt].

B Analysis [3 pts]

Write a report addressing each of the questions below. Please submit your report as a pdf file to NYU Classes.

In this section you will be exploring how different parameter choices affect the spectrogram of a sine sweep. As before, compute the sine sweep to be 1 second long sweeping between 500 Hz and 5 kHz with $a = 0.95$ at a sampling frequency $f_s = 44.1kHz$.

1. For each of the following variations, (i) compute and plot the time and frequency resolution of the resulting representation, (ii) discuss what happens as you introduce each variation, and (iii) discuss why these effects occur. Include all graphs in your report.

Unless otherwise stated, use Matlab's `spectrogram` function with the following default parameter values: window size = N, hop size = N/2, rectangular window, fft length = N, where N = 128.

(a) Default parameters (as above) [0.5 pt]

(b) window size = 256, 512, and 1024 [0.5 pt]

(c) h = N/4, h = N/16, h = N/32, where the window size N = 256 [0.5 pt]

(d) window type = hamming, window type = blackman [0.5 pt]

(e) fft length = 2N, 4N, 8N, where the window size N = 256 [0.5 pt]

(f) implementation = `spectrogram` and `my_spectrogram` [0.5 pt].